When Witnesses Defend: A Witness Graph Topological Layer for Adversarial Graph Learning

Naheed Anjum Arafat[1], **Debabrota Basu**[2], **Yulia Gel**[3], **Yuzhou Chen**[4]

[1]Nanyang Technological University, Singapore
[2]Univ. Lille, Inria, CNRS, Centrale Lille, UMR 9189 – CRIStAL, F-59000 Lille, France
[3]Virginia Tech, USA
[4]University of California, Riverside, USA
naheed_anjum@u.nus.edu, debabrota.basu@inria.fr, ygl@vt.edu, yuzhouc@ucr.edu

Abstract

Capitalizing on the intuitive premise that shape characteristics are more robust to perturbations, we bridge adversarial graph learning with the emerging tools from computational topology, namely, persistent homology representations of graphs. We introduce the concept of witness complex to adversarial analysis on graphs, which allows us to focus only on the salient shape characteristics of graphs, yielded by the subset of the most essential nodes (i.e., landmarks), with minimal loss of topological information on the whole graph. The remaining nodes are then used as witnesses, governing which higher-order graph substructures are incorporated into the learning process. Armed with the witness mechanism, we design *Witness Graph Topological Layer (WGTL)*, which systematically integrates both local and global topological graph feature representations, the impact of which is, in turn, automatically controlled by the robust regularized topological loss. Given the attacker's budget, we derive the important stability guarantees of both local and global topology encodings and the associated robust topological loss. We illustrate the versatility and efficiency of WGTL by its integration with five GNNs and three existing non-topological defense mechanisms. Our extensive experiments across six datasets demonstrate that WGTL boosts the robustness of GNNs across a range of perturbations and against a range of adversarial attacks.

1 Introduction

Recent studies has shown that Graph neural networks (GNNs) are vulnerable against adversarial attacks, i.e. small, often unnoticeable perturbations to the input graph might result in substantial degradation of GNN's performance in downstream tasks (Jin et al. 2021b). In turn, compared to non-graph data, adversarial analysis of graphs is still in its infancy (Sun et al. 2022). Hence, analysis of adversarial contagions and consequently development of robust GNN models able to withstand a wide spectrum of malicious attacks is of significant practical importance.

Presently, the three main strategies to defend GNNs against adversarial attacks are graph purification, adversarial training (Feng et al. 2019), and adversarial defense based neural architectures (Günnemann 2022; Mujkanovic et al. 2022). These existing methods largely rely on pairwise relationships in the graph at a node level while ignoring higher-order graph (sub)structures, their multi-scale properties, and interrelationships which are instrumental for the downstream learning task (Benson et al. 2018; Torres et al. 2021). Relying on pairwise relationships also results in the removal of a considerable amount of the edges that are actually clean edges, which decreases the gain in robustness (In et al. 2024).

In turn, in the last few years, we observe a spike of interest in the synergy of graph learning and Persistent Homology (PH) representations of graphs (Zhao and Wang 2019; Carrière et al. 2020; Horn et al. 2022; Yan et al. 2022; Chen, O'Bray, and Borgwardt 2022; Hajij et al. 2023; Chen and Gel 2023). PH representations enable us to glean intrinsic information about the inherent object shape. By shape here, we broadly understand properties which are invariant under continuous transformations such as twisting, bending, and stretching. This phenomenon can be explained by the important higher-order information, which PH-based shape descriptors deliver about the underlying graph-structured data. This leads to an enhanced GNN performance in a variety of downstream tasks, such as link prediction, node and graph classification (Hofer et al. 2020; Carrière et al. 2020; Chen, Coskunuzer, and Gel 2021; Yan et al. 2021; Horn et al. 2022). Furthermore, in view of the invariance with respect to continuous transformations, intuitively we can expect that shape characteristics are to yield higher robustness to random perturbations and adversarial attacks. While this intuitive premise of robustness and its relationship with DNN architectures has been confirmed by some recent studies (Chen, Coskunuzer, and Gel 2021; Gebhart, Schrater, and Hylton 2019; Goibert, Ricatte, and Dohmatob 2022), to the best of our knowledge, there are no works that attempt to incorporate PH-based graph representations for adversarial defense.

In this work, we bridge this gap by merging adversarial graph learning with PH representations of graph-structured data. Our key idea is to leverage the concept of witness complex for graph learning. This allows us firstly, to enhance the computational efficiency of the proposed topological defense, which is one of the primary bottlenecks for the wider adoption of topological methods, and lastly, to reduce the impact of less important or noisy graph information. In particular, the goal of the witness complex is to accurately estimate intrinsic shape properties of the graph using not all available graph information, but *only* a subset of the most representative nodes, called *landmarks*. The remaining nodes are then used as *witnesses*, governing which higher-order graph substructures shall be incorporated into the process of extracting shape characteristics and the associated graph learning task. This mechanism naturally results in two main benefits. First, it allows us to drastically reduce the computational costs. Second, it allows us to extract salient shape characteristics (i.e., skeleton shape). Our topological defense takes a form of the *Witness Graph Topological Layer (WGTL)* with three novel components: *local and global witness complex-based topological encoding*, *topology prior aggregation*, and *robustness-inducing topological loss*.

The *local witness complex-based features* encapsulate graph topology within the local node neighborhoods, while the *global witness complex-based features* describes global graph topology. Using only local topology prior to the loss function might be vulnerable to local attacks, while only global topology prior might be more susceptible to global attacks. To defend against both types of attacks, both local and global topology prior needs to be combined, thus motivating the design of the topology prior aggregator. Inspired by works such as Hu et al. (2019); Carriere et al. (2021), we *use the robust topological loss as a regularizer to a supervised loss for adversarially robust node representation learning*. This allows to control which shape features are to be included into the defense mechanism.

Our Contributions.

- We propose the first approach which systematically bridges adversarial graph learning with persistent homology representations of graphs.
- We introduce a novel topological adversarial defense for graph learning, i.e. the *Witness Graph Topological Layer (WGTL)*, based on the notion of the witness complex. WGTL systematically integrates both local and global higher-order graph characteristics. Witness complex enables us to focus only on the salient shape characteristics delivered by the landmark nodes, thereby reducing the computational costs and minimizing the impact of noisy graph information.
- We derive the stability guarantees of both local and global topology encodings and the robust topological loss, given an attacker's budget. These guarantees show that local and global encodings are stable to external perturbations, while the stability depends on the goodness of the witness complex construction.
- Our extensive experiments spanning six datasets and eight GNNs indicate that WGTL boosts robustness capabilities of GNNs across a wide range of local and global adversarial attacks, resulting in relative gains up to 18%. WGTL also smoothly integrates with other existing defenses, such as Pro-GNN, GNNGuard and SimP-GCN improving the relative performance up to 4.95%, 15.67% and 5.7% respectively.

Existing Defenses for GNNs. There are broadly three types of defenses: graph purification-based, adversarially robust training and adversarially robust architecture (Günnemann 2022). Notable defenses that purify the input graph include SG-GSR (In et al. 2024), Pro-GNN (Jin et al. 2020) and SVD-GCN (Entezari et al. 2020). These methods learns to remove adversarial edges from the poisoned graph without considering higher-order interactions. In contrast, WGTL primarily focuses on learning the key higher-order graph interactions at both local and global levels and then adaptively assessing their potential defense role via topological regularizer. The local and global topological encodings remain robust despite the false positive edges; as a result, WGTL alleviates the problems associated with false positive edges (In et al. 2024), enhancing the overall resilience against attacks. The adversarial training-based defense methods augment node features with gradients (Kong et al. 2020), or datasets by generating worst-case perturbations (Xu et al. 2019). The goal is to train with the worst-case adversarial perturbations such that the learnt model weights become more robust against worst-case perturbation (Günnemann 2022). However, adversarial training can not defend against more severe perturbation than the ones they were trained with. Better architectures such as VAE (Zhang and Ma 2020), Bayesian uncertainty quantification (Feng, Wang, and Ding 2021), and Attention (Zhu et al. 2019; Tang et al. 2020) have also been proposed for adversarial defense. However, none of these tools have explored the use of robust, graph topological features as prior knowledge for improved defense. Recently, Gabrielsson et al. (2020) designed a topology-driven attack on images and topological loss, but this approach does not consider graph data and no adversarial defense is proposed. Among topology-driven defenses, GNNGuard (Zhang and Zitnik 2020) considers graphlet degree vectors to encode node structural properties such as triangles and betweenness centrality. However, unlike the PH features used in WGTL, the graphlet approach is empirical, without theoretically guaranteed robustness properties. The use of a robust loss function as a regularizer for defense is not new; for instance, Zügner and Günnemann (2019b) proposed *robust hinge loss* for defense. However, it is unknown if topological losses (Hu et al. 2019; Gabrielsson et al. 2020) can improve adversarial robustness or not.

2 Background: Graphs, Persistent Homology, Complexes, Adversarial ML

Topology of Graphs. $\mathcal{G} \triangleq (\mathcal{V}, \mathcal{E}, \boldsymbol{X})$ denotes an attributed graph. $\mathcal{V}$ is a set of N nodes. $\mathcal{E}$ is a set of edges. $\boldsymbol{X} \in \mathbb{R}^{N\times F}$ is a node feature matrix, where each node corresponds to an F dimensional feature. The adjacency matrix of $\mathcal{G}$ is a symmetric matrix $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ such that $\boldsymbol{A}_{uv} \triangleq \omega_{uv}$, i.e., edge weight, if nodes u and v are connected and 0, otherwise. For unweighted graphs, we observe $\omega_{uv} = 1$. Furthermore, $\boldsymbol{D}$ represents the degree matrix of $\mathcal{G}$, such that $\boldsymbol{D}_{uu} \triangleq \sum_{v\in\mathcal{V}} \boldsymbol{A}_{uv}$ and 0, otherwise.

The central ideas leveraged in this paper are the local and global topology of a graph. The topology of a graph is defined by corresponding geodesic distance. The geodesic distance $d_{\mathcal{G}}(u, v)$ between a pair of vertices u and $v \in \mathcal{V}$ is defined as the length of the shortest path between u and v. The path length is defined as the sum of weights of the edges connecting the vertices u and v. Endowed with the canonical metric induced by the geodesic distance $d_{\mathcal{G}} : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}^{\geq 0}$, a weighted simple graph $\mathcal{G}$ transforms into a metric space $(\mathcal{V}, d_{\mathcal{G}})$. For a given positive real number $\epsilon > 0$, the set of nodes that are no more than geodesic ϵ away from a given node determines the local topology of that node. When $\epsilon = \mathrm{Diam}(\mathcal{G})$, i.e. the diameter of $\mathcal{G}$, we retrieve the global topology of the graph. Increasing ϵ from 1 to $\mathrm{Diam}(\mathcal{G})$ allows us to retrieve the evolution of the inherent graph features, like connected components, cycles, voids, etc. (Edelsbrunner, Letscher, and Zomorodian 2002; Zomorodian 2005).

Persistent Homology. In order to study the evolution of graph features, we take a Persistent Homology (PH)-based approach. Persistent homology is a method of computational topology that quantifies topological features by constructing simplicial complexes, i.e. a generalised graph with higher-order connectivity information such as cliques, over the dataset. For example, a unweighted subgraph of $\mathcal{G}$, say $\mathcal{G}_\alpha$, consisting of only edges with length more than α is a simplicial complex. The d-th homology group of a simplicial complex $\mathcal{G}_\alpha$ consists of its d-dimensional topological features, such as connected components ($d = 0$), cycles ($d = 1$), and voids ($d = 2$). Now, as we increase α, we observe that more and more edges are removed from $\mathcal{G}$. Thus, we obtain a nested sequence of simplicial complexes $\mathcal{G}_{\alpha_1} \subseteq \ldots \subseteq \mathcal{G}_{\alpha_n} = \mathcal{G}$ for $\alpha_1 \leq \alpha_2 \leq \ldots \leq \alpha_n$. This nested sequence of simplicial complexes is called a *graph filtration* and α_i's denote the filtration values. To make the process more systematic and informative, often an abstract simplicial complex $\mathscr{K}(\mathcal{G}_{\alpha_j})$ is constructed on each $\mathcal{G}_{\alpha_j}$, resulting in a *filtration* of complexes $\mathscr{K}(\mathcal{G}_{\alpha_1}) \subseteq \ldots \subseteq \mathscr{K}(\mathcal{G}_{\alpha_n})$. For a more detailed discussion on graph filtration, we refer to Hofer et al. (2020).

The key idea of PH is to choose multiple scale parameters α and study changes in topological features that occur to $\mathcal{G}$, which evolves with respect to α. Equipped with the filtration of complexes, we can trace data shape patterns, i.e. the d homology groups, such as independent components, holes, and cavities which appear and merge as scale α changes. For each topological feature ρ, we record the indices b_ρ and d_ρ of $\mathscr{K}(\mathcal{G}_{b_\rho})$ and $\mathscr{K}(\mathcal{G}_{d_\rho})$, where ρ is first and last observed, respectively. We say that a pair (b_ρ, d_ρ) represents the birth and death times of ρ, and $(d_\rho - b_\rho)$ is its corresponding lifespan (or *persistence*). In general, topological features with longer persistence are considered valuable, while features with shorter persistence are often associated with topological noise. The extracted topological information over the filtration $\{\mathscr{K}_{\alpha_j}\}$ is then represented in $\mathbb{R}^2$ as a *Persistence Diagram (PD)*, such that $\mathrm{PD} = \{(b_\rho, d_\rho) \in \mathbb{R}^2 : d_\rho > b_\rho\} \cup \Delta$. $\Delta = \{(t,t) | t \in \mathbb{R}\}$ is the diagonal set containing points counted with infinite multiplicity. Another useful representation of persistent topological features is *Persistence Image* (PI) that vectorizes the persistence diagram with a Gaussian kernel and a piece-wise linear weighting function (Adams et al. 2017). Persistence images are deployed to make a classifier "topology-aware" and are known to be helpful in graph classification (Zhao and Wang 2019; Rieck et al. 2020).

Witness Complexes. There are multiple ways to construct an abstract simplicial complex $\mathscr{K}$ (Zomorodian 2005). Due to its computational benefits, one of the widely adopted approaches is a *Vietoris-Rips complex* (VR). However, the VR complex uses the entire observed data to describe the underlying topological space, and thus, does not efficiently scale to large and noisy datasets (Zomorodian 2010). In contrast, a *witness complex* captures the data shapes using only on a significantly smaller subset $\mathfrak{L} \subseteq \mathcal{V}$, called a set of *landmarks* (De Silva and Carlsson 2004). In turn, all other points in $\mathcal{V}$ are used as "witnesses" that govern the appearances of simplices in the witness complex. Arafat, Basu, and Bressan (2020) demonstrate algorithms to construct landmark sets, their computational efficiencies, and stability of the induced *witness complex*. We leverage witness complex to scale to large graph datasets.

Definition 2.1 (Weak Witness Complex (De Silva and Carlsson 2004))**.** We call $w \in \mathcal{V}$ to be a *weak witness* for a simplex $\sigma = [v_0 v_1 \ldots v_l]$, where $v_i \in \mathcal{V}$ for $i = 0, 1, \ldots, l$ and $l \in \mathbb{N}$, with respect to $\mathfrak{L}$ if and only if $d_{\mathcal{G}}(w, v) \leq d_{\mathcal{G}}(w, u)$ for all $v \in \sigma$ and $u \in \mathfrak{L} \setminus \sigma$. The *weak witness complex* $\mathrm{Wit}(\mathfrak{L}, \mathcal{G})$ of the graph $\mathcal{G}$ with respect to the landmark set $\mathfrak{L}$ has a node set formed by the landmark points in $\mathfrak{L}$, and a subset σ of $\mathfrak{L}$ is in $\mathrm{Wit}(\mathfrak{L}, \mathcal{G})$ if and only if there exists a corresponding weak witness in the graph $\mathcal{G}$.

Adversarial ML and Robust Representations. Graph Neural Networks (GNNs) aim to learn a labelling function that looks into the features the nodes in the graph $\mathcal{G}$ and assign one of the C labels $y_v \in \{1, \ldots, C\}$ to each node $v \in \mathcal{V}$ (Kipf and Welling 2016). In order to learn to labelling, GNNs often learn compact, low-dimensional representations, aka *embeddings* $R : \mathcal{G} \times \mathcal{V} \to \mathbb{R}^r$, for nodes that capture the structure of the nodes' neighbourhoods and their features, and then apply a classification rule $f : \mathbb{R}^r \to \{1, \ldots, C\}$ on the embedding (Kipf and Welling 2016; Hamilton, Ying, and Leskovec 2017; Zhang and Zitnik 2020).

The goal of a robust GNN training mechanism is to learn a labelling function $f \circ R$ such that the change in the predicted labels, i.e. $|(f \circ R)(\mathcal{G}') - (f \circ R)(\mathcal{G})|$, is the minimum, when a graph $\mathcal{G}$ is adversarially perturbed to become $\mathcal{G}'$ (Zhang and Zitnik 2020). The budget of perturbation is defined by $\delta = \|\mathcal{G} - \mathcal{G}'\|_p$. p is often fixed to 0 or 1 (Xu et al. 2019; Wu et al. 2019b; Zügner and Günnemann 2019b). There are different ways to design a robust training mechanism, such as training with an adversarially robust loss function (Xu et al. 2019), using a stabilising regularizer to the classification loss (Zügner and Günnemann 2019b), learning a robust representation of the graph (Engstrom et al. 2019; Liu et al. 2023), etc.

In this paper, we aim to design a *robust representation* R of the graph $\mathcal{G}$ using its persistent homologies. Specifically, *we call a graph representation R robust,* if for $p, q > 0$,

$$\|R(\mathcal{G}) - R(\mathcal{G}')\|_p = \mathcal{O}(\delta), \text{ when } \|\mathcal{G} - \mathcal{G}'\|_q = \delta.$$

In the following section, we propose WGTL, which is a topology-aware graph representation, and show that WGTL achieves this robust representation property.

3 Learning a Robust Topology-aware Graph Representation

The general idea is that encoding robust graph structural features as prior knowledge to a graph representation learning framework should induce a degree of robustness against adversarial attacks. Graph measures that capture global properties of the graph and measures that rely on aggregated statistics are known to be robust against small perturbations (Borgatti, Carley, and Krackhardt 2006). Examples include degree distribution, clustering coefficients, average path length, diameter, largest eigenvalue and the corresponding eigenvector, certain centrality measures, e.g., betweenness and closeness centralities. However, these measures are not multiscale in nature. Therefore, they fail to encapsulate global graph structure at multiple levels of granularity. Many of them, e.g., degree distribution, clustering coefficients, only encode 1-hop or 2-hop information. Such information can be learned by a shallow GNN through message passing, rendering such features less useful as a prior. Features such as average path length and diameter are too coarse-scale (scalar-valued) and do not help a GNN to discern the nodes. Since existing robust graph features can not encode both local and global topological information at multiple scales, we introduce local and global topology encodings based on persistent homology as representations to the GNNs (Section 3.1). We also propose to use a topological loss as regularizer to learn topological features better (Section 3.2).

3.1 Witness Graph Topological Layer (WGTL)

Component I: Local Topology Encoding. *Local topology encoding* component of WGTL (Figure **??**) computes local topological features of every node in three steps. First, we choose a landmark set $\mathfrak{L}$ from the input graph $\mathcal{G}$. An important hyperparameter of the local topology encoding is the choice of the number of landmarks. Choosing too few landmarks would reduce the informativeness of the latent embedding. Choosing too many landmarks (i.e., $|\mathcal{V}|$), on top of being computationally expensive, might be redundant because the topological features of a neighboring node are likely to be the same. Secondly, we use the landmarks to construct an ϵ-net of $\mathcal{G}$ (Arafat, Basu, and Bressan 2020), i.e. a set of subgraphs $\{\mathcal{G}_l^\epsilon\}_{l\in\mathfrak{L}}$. Here, $\epsilon \triangleq \max_{l_1,l_2\in\mathfrak{L}} 0.5 d_{\mathcal{G}}(l_1,l_2)$. We compute witness complex for each of these $\mathcal{G}_l^\epsilon$'s, and the corresponding persistence images $\mathrm{PI}(\mathrm{Wit}(\mathcal{G}_l^\epsilon))$. Finally, we attribute the PIs of the landmarks to each node in its ϵ-cover and pass them through a vision transformer model to compute the local topology encoding, i.e. $\boldsymbol{Z}_{T_L} = \mathrm{Transformer}(\mathrm{PI}(\mathrm{Wit}(\mathcal{G}^\epsilon))_1, \ldots, \mathrm{PI}(\mathrm{Wit}(\mathcal{G}^\epsilon))_N)$. The local topology encoding $\boldsymbol{Z}_{T_L}$ is a latent embedding of local topological features of each node in $\mathcal{G}$.

When the attack model poisons the adjacency matrix, especially in the cases of global attacks, the local topological encodings are also implicitly perturbed. In Theorem 3.1, we show that local topological encodings are stable w.r.t. perturbations in the input graph. Specifically, if an attacker's budget is $\mathcal{O}(\delta)$, the encoded local topology is perturbed by $\mathcal{O}(C_\epsilon(\delta+\epsilon))$. The bound indicates the trade-off due to landmark selection. If we select less landmarks, computation becomes faster and we encode topological features of a larger neighborhood. But increase in C_ϵ yields less stable encoding. Whereas if we select more landmarks, we get more stable encoding but we loose informativeness of the local region and computational efficiency.

Theorem 3.1 (Stability of the encoded local topology)**.** *Let us denote the persistence diagram obtained from local topology encoding of $\mathcal{G}$ as $\mathrm{T}(\mathcal{G})$ (Figure **??**). For any $p < \infty$ and C_ϵ being the maximum cardinality of the ϵ-neighborhood created by the landmarks, we obtain that for any graph perturbation $\|\mathcal{G}-\mathcal{G}'\|_1 = \mathcal{O}(\delta)$ the final persistence diagram representation changes by $W_p(\mathrm{T}(\mathcal{G}), \mathrm{T}(\mathcal{G}')) = \mathcal{O}(C_\epsilon \delta)$, if we have access to Cěch simplicial complexes, and $W_p(\mathrm{T}(\mathcal{G}), \mathrm{T}(\mathcal{G}')) = \mathcal{O}(C_\epsilon(\delta+\epsilon))$, if Witness complex is used for the Local Persistence Images.*

Component II: Graph Representation Learning. The component II of WGTL deploys in cascade M GNN layers with ReLU activation function and weights $\{\boldsymbol{\Theta}^{(m)}\}_{m=1}^M$. The representation learned at the m-th layer is given by $\boldsymbol{Z}_G^{(m+1)} = \mathrm{ReLU}(\widetilde{\boldsymbol{D}}^{-\frac{1}{2}} \widetilde{\boldsymbol{A}} \widetilde{\boldsymbol{D}}^{\frac{1}{2}} \boldsymbol{Z}_G^{(m)} \boldsymbol{\Theta}^{(m)})$. Here, $Z_G^{(0)} = \mathcal{G}$, $\widetilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$, and $\widetilde{\boldsymbol{D}}$ is the corresponding degree matrix.

Component III: Global Topology Encoding. The *global topological encoding* represents the global witness complex-based topological features of a graph (Component III in Figure **??**). First, we use the input adjacency matrix to compute the lengths of all-pair shortest paths (geodesics) among the nodes. The topological space represented by the geodesic distance matrix is used to compute the global witness complex-based persistence image $\mathrm{PI}(\mathrm{Wit}(\mathcal{G}))$ of the graph (Arafat, Basu, and Bressan 2020). Finally, the persistence image representation is encoded by a Convolutional Neural Network (CNN)-based model to obtain the *global topological encoding* $\boldsymbol{Z}_{T_G} \triangleq \xi_{\text{max}}(\mathrm{CNN}(\mathrm{PI}(\mathrm{Wit}(\mathcal{G}))))$. Here, $\xi_{\text{max}}(\cdot)$ denotes global max-pooling operation. The global topology encoding encapsulates the global topological features, such as equivalent class of connected nodes, cycles and voids in the graph.

The stability of global persistence diagram representation is a well-known classical result in persistence homology (Cohen-Steiner, Edelsbrunner, and Harer 2005; Chazal et al. 2008). However, given an attacker's budget of δ, the stability of the encoded global topology is an important result for the practical purposes of this paper. Theorem 3.2 shows that under a $\mathcal{O}(\delta)$ perturbation of the input graph, the global topology encoding is perturbed by $\mathcal{O}(\delta+\epsilon)$. Hence, the global topological encoding inherits the robustness property of persistent homology and induces robust learning under adversarial attacks.

Proposition 3.2 (Stability of the encoded global topology)**.** *If the landmarks selected for the witness complex induce an ϵ-net of the graph with $\epsilon > 0$, we obtain that for any graph perturbation $\|\mathcal{G}-\mathcal{G}'\|_1 = \mathcal{O}(\delta)$ the global persistence image representation changes by $\|\mathrm{PI}(\mathrm{Wit}^{\mathrm{glob}}(\mathcal{G})) - \mathrm{PI}(\mathrm{Wit}^{\mathrm{glob}}(\mathcal{G}'))\|_\infty = \mathcal{O}(\delta+\epsilon)$, and it reduces to $\mathcal{O}(\delta)$, if we have access to the Cěch simplicial complexes for $\mathcal{G}$.*

WGTL: Aggregating Global and Local Encodings. We can aggregate the local and global topology encodings with the latent embedding of graph convolution layers in different ways. Figure **??** shows the approach that empirically provides the most effective defense against adversarial attacks.

The aggregation of the three encodings is computed in two steps. First, to adaptively learn the intrinsic dependencies between learnt node embedding and latent local topological encodings, we utilize the attention mechanism to focus on the importance of task relevant components in the learnt representations, i.e. $(\alpha_G, \alpha_{T_L}) \triangleq \text{Att}(\boldsymbol{Z}_H, \boldsymbol{Z}_{T_L})$. In practice, we compute attention coefficients as $\alpha_i = \text{softmax}_i(\Upsilon_{\text{Att}} \tanh(\Xi \boldsymbol{Z}_i))$, where $\Upsilon_{\text{Att}} \in \mathbb{R}^{1\times d_{\text{out}}}$ is a linear transformation, Ξ is the trainable weight matrix, and the softmax function is used to normalize the attention vector. Then, we obtain the final embedding by combining two embeddings $\boldsymbol{Z}_{\text{AGG}} = \alpha_G \times \boldsymbol{Z}_G + \alpha_{T_L} \times \boldsymbol{Z}_{T_L}$. Finally, we combine the learnt embedding $\boldsymbol{Z}_{\text{AGG}}$ with the latent global topological representation $\boldsymbol{Z}_{T_G}$, such that $\boldsymbol{Z}_{\text{WGTL}} = \boldsymbol{Z}_{\text{AGG}}\boldsymbol{Z}_{T_G}$. The node representation $\boldsymbol{Z}_{\text{WGTL}}$ encapsulates both global and local topology priors. We call $\boldsymbol{Z}_{\text{WGTL}}$ the *aggregated topological priors*. We feed $\boldsymbol{Z}_{\text{WGTL}}$ into a graph convolutional layer and use a differentiable classifier (here we use a softmax layer) to make node classification. In the following, we show the stability of the aggregated topological priors.

Proposition 3.3 (Stability of the aggregated topological encoding)**.** *If the landmarks selected for the witness complex induce an ϵ-net of the graph with $\epsilon > 0$ and L_{GNN} is the Lipschitz constant of the GNNs in Component II, then for a perturbation $\|\mathcal{G} - \mathcal{G}'\|_1 = \mathcal{O}(\delta)$, the encoding $\boldsymbol{Z}_{\text{WGTL}}$ changes by*

$$\|\boldsymbol{Z}_{\text{WGTL}}(\mathcal{G}) - \boldsymbol{Z}_{\text{WGTL}}(\mathcal{G}')\|_1 = \mathcal{O}((C_\epsilon + L_{\text{GNN}})(\delta + \epsilon)^2).$$

Proposition 3.3 shows that the final representations computed by WGTL is stable under adversarial attacks. The stability depends on the approximation trade-off induced by the landmark set and the Lipschitz stability of the GNN layers (Jia et al. 2023).

3.2 Topological Loss as a Regularizer

In Section 3.1, we propose using the aggregated topology encodings to predict node labels for downstream node classification tasks through a GNN backbone. In this case, we use a supervised loss L_{supv} that facilitate learning the aggregated topology priors for classification.

However, the supervised loss function only explicitly enforces misclassification constraints on the defense model. It does not explicitly enforce any topological constraint such that the topological encodings themselves iteratively become more robust while training. Hence, for increased robustness, we propose to use topological loss L_{topo} that explicitly encodes the birth and death of the topological features in the auxiliary graph (ref. Figure **??**) reconstructed from the transformer output. Specifically,

$$L_{topo,k}(\text{T}(\mathcal{G})) \triangleq \sum_{i=1}^{m}(d_i - b_i)^p \left(\frac{d_i + b_i}{2}\right)^q, \qquad (1)$$

where m is the number of points in the persistence diagram of the auxiliary graph reconstructed from the transformer output and $k = \max\{p, q\}$. In practice, we use $k = 2$. Use of such topological loss was first proposed for image segmentation (Hu et al. 2019). Gabrielsson et al. (2020) uses it as a regularizer in designing GAN and adversarial attacks on images. In contrast, we use it to induce stability in the encoding and to defend against adversarial attacks. The benefits of using the topological loss are two-fold:

(i) **Persistent and Stable Feature Selection:** Minimising $L_{topo,k}$ causes removal of topological features with smaller persistence, i.e., $(d_i - b_i)$. As such, the regularizer acts as a sparsity-inducing feature selector. By minimising L_{topo}, we are training to learn latent representation such that only the most persistent features remain in the encoded local topology. Such features are known to be more stable and represent more robust structures of the graph.

(ii) **Robustness to Local Perturbations:** A localized attack perturbing certain nodes or edges is expected to appear as topological noise in the final persistent diagram, and should exhibit lower persistence. Since minimizing L_{topo} forces the local topology encodings to eliminate features with small persistences, using L_{topo} as a regularizer with L_{supv} induces robustness to local perturbations in final classification tasks.

Proposition 3.4 quantifies the stability of the topological regularizer $L_{topo,k}$ under any attack with perturbation budget $\mathcal{O}(\delta)$. Specifically, it shows that the stability depends on a trade-off between the maximum persistence of the final graph representation, $A_\Phi(\mathcal{G})$, in Figure **??**, and the number of non-zero persistent features in the final encoding. Hence, it reflects our discussion above.

Proposition 3.4 (Stability of L_{topo})**.** *Let us assume that the cardinality of any ϵ-neighborhood of $\mathcal{G}$ grows polynomially, i.e. $C_\epsilon = \mathcal{O}(\epsilon^{-M})$ for an $M > 0$. If m is the number of points in the persistence diagram, $2k = 2\max\{p, q\} > M$, and $A(\mathcal{G})$ is the auxiliary graph constructed from the local topology encodings (Fig. **??**), $L_{topo,k}(\text{T}(\mathcal{G}))$ is stable w.r.t. a perturbation of $\mathcal{G}$, i.e. $\|\mathcal{G} - \mathcal{G}'\|_1 = \delta$.*

$$\begin{aligned}&\left|L_{topo,k}(\text{T}(\mathcal{G})) - L_{topo,k}(\text{T}(\mathcal{G}'))\right| \\ &= \mathcal{O}\left(\left(\epsilon^{-4kM}\text{Diam}(A(\mathcal{G})) + m\epsilon^{-2k}\text{Diam}(\mathcal{G})^{2k}\right)\delta\right).\end{aligned}$$

4 Experimental Evaluation

We evaluate the proposed WGTL on the node classification task for clean and attacked graphs across a range of perturbation rates. We validate the proposed approach on six benchmark datasets: Citeseer, Cora, Pubmed, Polblogs, OGBN-Arxiv and Snap-patents. *We report mean and standard deviation of accuracies over* 10 *runs. The best performance is highlighted in bold while the best result on a dataset for a given perturbation rate is indicated by* *. Note that, throughout our experiments, we use 0-dimensional topological features. All the hyperparameters are chosen by performing cross-validation. We defer the dataset descriptions, implementation details, ablation studies, impact of the #landmarks on performance, comparison with Vietoris-Rips and additional experimental results such as handling node features, heterophilic graphs, adaptive attacks and the adoption of other topological vectorization methods in WGTL to the Appendix section in the extended paper (Arafat et al. 2024).

Landmark Selection for Local and Global Topology Encodings. There are several approaches to selecting landmarks, e.g., random selection (De Silva and Carlsson 2004), maxmin selection (De Silva and Carlsson 2004), ϵ-net (Arafat, Basu, and Bressan 2020) based and centrality-based selection (Chen and Gel 2023). In our experiments, we select landmarks based on degree centrality. As shown by Chen and Gel (2023), doing so helps to improve the classification performance. On Cora-ML, Citeseer and Polblogs, we select

Table 1: Comparison of performances (avg. accuracy±std.) with existing defenses under mettack.

Dataset	Models	Perturbation Rate		
		0%	5%	10%
Cora-ML	Pro-GNN	82.98±0.23	80.14±1.34	71.59±1.33
	Pro-GNN+WGTL	**83.85±0.38**	**81.90±0.73**	**72.51±0.76**
	GCN+GNNGuard	83.21±0.34	76.57±0.50	69.13±0.77
	GCN+GNNGuard+WGTL	***84.78±0.43**	***83.23±0.82**	***79.96±0.49**
	SimP-GCN	79.52±1.81	74.75±1.40	70.87±1.70
	SimP-GCN+WGTL	**81.49±0.52**	**76.65±0.65**	**72.88±0.83**
Citeseer	ProGNN	72.34±0.99	68.96±0.67	67.36±1.12
	ProGNN+WGTL	**72.83±0.94**	**71.85±0.74**	**70.70±0.57**
	GCN+GNNGuard	71.82±0.43	70.79±0.22	66.86±0.54
	GCN+GNNGuard+WGTL	**73.37±0.63**	**72.57±0.17**	**66.93±0.21**
	SimP-GCN	73.73±1.54	73.06±2.09	72.51±1.25
	SimP-GCN+WGTL	***74.32±0.19**	***74.05±0.71**	***73.09±0.50**
Pubmed	Pro-GNN	87.33±0.18	87.25±0.09	87.20±0.12
	Pro-GNN + WGTL (ours)	**87.90±0.30**	***87.77±0.08**	***87.67±0.22**
	GCN+GNNGuard	83.63±0.08	79.02±0.14	76.58±0.16
	GCN+GNNGuard+WGTL	OOM	OOM	OOM
	SimP-GCN	*88.11±0.10	86.98±0.19	86.30±0.28
	SimP-GCN+WGTL	OOM	OOM	OOM
Polblogs	GCN+GNNGuard	95.03±0.25	73.25 ±0.16	72.76±0.75
	GCN+GNNGuard+WGTL	***96.22±0.25**	***73.62±0.22**	***73.72±1.00**
	SimP-GCN	89.78±6.47	65.75±5.03	61.53±6.41
	SimP-GCN+WGTL	**94.56±0.24**	**69.78±4.10**	**69.55±4.42**

5% nodes, on Pubmed and Snap-patents we select 2% nodes and on OGBN-arxiv we select 0.05% nodes as landmarks to keep #landmarks roughly invariant across datasets.

Adversarial Attacks: Local and Global. We deploy four local and global poisoning attacks, with perturbation rates, i.e., the ratio of changed edges, from 0% to 10%, to evaluate the robustness of WGTL. We consider a fixed GCN without weight re-training as the surrogate for all attacks. As a local attack, we deploy nettack (Zügner, Akbarnejad, and Günnemann 2018). *Due to the stability of WGTL and topological regularizer, we expect to be robust to such local attacks.* As global (non-targeted) poisoning attacks, we deploy mettack (Zügner and Günnemann 2019a), and two topological attacks, namely PGD (Xu et al. 2019) and Meta-PGD (Mujkanovic et al. 2022). Mettack treats the graph as a hyperparameter and greedily selects perturbations based on meta-gradient for node pairs until the budget is exhausted. We keep all the default parameter settings (e.g., $\lambda = 0$) following the original implementation (Zügner and Günnemann 2019a). For Cora-ML, Citeseer and Polblogs, we apply the most effective Meta-Self variant, while for Pubmed, we apply the approximate variant (A-Meta-Self) to save memory and time (Jin et al. 2020). *Though global attacks are expected to be more challenging while using topological features, we demonstrate that WGTL still yields significant robustness.* Further details on attack implementations and attackers' budgets are discussed in the extended paper (Arafat et al. 2024), so are the results for PGD and Meta-PGD attacks.

Objectives. *We implemented and compared WGTL with 3 existing defenses and 5 GNN backbones to study five questions. (Q1) Can WGTL enhance the robustness of the existing defenses? (Q2) Can WGTL enhance the robustness of existing backbone graph convolution layers? (Q3) Is WGTL still effective when the topological features are computed on poisoned graphs instead of clean graph? (Q4) How WGTLs performs on large graphs? (Q5) Is WGTL computationally efficient?*

Q1. Performance of WGTL Defense w.r.t. Existing Defenses. We compare our method with three state-of-the-art defenses: Pro-GNN (Jin et al. 2020), GNNGuard (Zhang and Zitnik 2020), and SimP-GCN (Jin et al. 2021a). Table 1 illustrates the comparative performances on three citation networks under aglobal attack, i.e. mettack. We observe that our Pro-GNN+WGTL is always better than other baselines on all datasets. Following Jin et al. (2020), we omit Pro-GNN for Polblogs. As a consequence, we gain 0.68% - 4.96% of relative improvements on Cora-ML and Citeseer. Similarly, we observe that GCN+GNNGuard+WGTL outperforms GCN+GNNGuard and SimP-GCN+WGTL outperforms Simp-GCN by 0.10% - 15.67% and 2.4% - 5.7%, respectively, across all datasets. The results reveal that WGTL enhances not only model expressiveness but also the robustness of the GNN-based models. The performance comparison under nettack are in the extended paper (Arafat et al. 2024).

Q2. WGTL Enhances Robustness of GNNs. WGTL is flexible in the sense that it can employ existing GNN layers to enhance their robustness. To be precise, we have employed the existing GNN backbones as component II in Figure **??** to enhance their robustness. Since global attacks target global graph topology, global poisoning attacks are supposed to be more challenging for the proposed *topology-based* defense WGTL. Despite that, we observe that WGTL consistently improves the robustness of all backbone GNNs in Table 2. The performance of our method, including that of the backbone GNNs, deteriorates faster on Polblogs than on the other datasets. This is because Polblogs does not have node features, and having informative node features can help GNN to differentiate between nodes and to learn meaningful representations despite changes in the graph structure. With node features lacking, the Polblogs has comparatively less resilience against graph structural perturbations. The results with the SGC backbone (Wu et al. 2019a) are in the Appendix section of the extended paper (Arafat et al. 2024).

Q3. Performance of WGTL on Poisoned Graphs. So far, the local and global topological features are computed on clean graphs assuming that these features can be computed before attacker poisons the graph. However, such assumption is restrictive as the attacker might poison the graph at any point before and during training. As a result, the topological features computed by WGTL might also be poisoned, as they were computed based on the poisoned graph. WGTL$_{\text{P}}$ employs poisoned graphs as inputs in the schematics of Figure **??** and **??**. We present the performance of WGTL$_{\text{P}}$ on Cora-ML and Polblogs under mettack in Table 3. We observe a consistent improvement over the baseline models across various datasets and perturbation rates. In this setting, we find GAT+WGTL$_{\text{P}}$ and GraphSAGE+WGTL$_{\text{P}}$ to be the best performing models. We observe that *WGTL robustifies the existing backbones, e.g., GAT and SAGE, more compared to all other defenses.*

Q4. Performance of WGTL on large-scale graph. We have applied PRBCD attack to generate the perturbed OGBN-arXiv graph since we found that other attacks, such as

Table 2: Robustness of various backbone GNNs (avg. accuracy±std.) under mettack

Dataset	Model	Perturbation Rate		
		0%	5%	10%
Cora-ML	GCN	82.87±0.83	76.55±0.79	70.39±1.28
	GCN + WGTL (ours)	**83.83±0.55**	**78.63±0.76**	**73.41±0.82**
	ChebNet	80.74±0.42	74.35±1.2	66.62±1.44
	ChebNet + WGTL (ours)	**82.96±1.08**	**76.00±1.22**	**69.49±0.89**
	GAT	84.25±0.67	79.88±1.09	72.63±1.56
	GAT + WGTL (ours)	***86.07±2.10**	**80.80±0.87**	**75.80±0.79**
	GraphSAGE	81.00±0.27	74.81±1.2	70.92±1.18
	GraphSAGE + WGTL (ours)	**83.63±0.35**	***82.61±0.65**	***81.19±1.13**
Polblogs	GCN	94.40±1.47	71.41±2.42	69.16±1.86
	GCN + WGTL (ours)	***95.95±0.15**	**74.62±0.42**	**72.84±0.86**
	ChebNet	73.10±7.13	67.63±1.71	67.36±0.85
	ChebNet + WGTL (ours)	**92.50±1.10**	**71.17±0.10**	**68.03±0.87**
	GAT	95.28±0.51	75.83±0.90	73.11±1.20
	GAT + WGTL (ours)	**95.87±0.26**	***83.13±0.32**	**80.06±0.50**
	GraphSAGE	94.52±0.27	77.44 ± 1.71	74.66±0.85
	GraphSAGE + WGTL (ours)	**95.58±0.50**	**82.62±0.65**	***81.49±0.86**

Table 3: Performance on poisoned graph (avg. accuracy±std) under mettack.

Dataset	Models	Perturbation Rate		
		0%	5%	10%
Cora-ML	GCN	82.87±0.83	76.55±0.79	70.39±1.28
	GCN+WGTL$_\text{P}$(ours)	**83.83±0.55**	**76.96±0.76**	**71.31±0.85**
	GAT	84.25±0.67	79.88±1.09	72.63±1.56
	GAT+WGTL$_\text{P}$(ours)	***86.07±2.10**	**81.43±0.75**	***73.74±1.92**
	GraphSAGE	81.00±0.27	74.81±1.20	70.92±1.18
	GraphSAGE+WGTL$_\text{P}$(ours)	**83.63±0.35**	***82.15±1.25**	**73.57±0.73**
	ProGNN	82.98±0.23	80.14±1.34	71.59±1.33
	ProGNN+WGTL$_\text{P}$(ours)	**83.85±0.38**	**81.69±1.83**	**72.71±1.26**
	GCN+GNNGuard	83.21±0.34	76.57±0.50	69.13±0.77
	GCN+GNNGuard+WGTL$_\text{P}$(ours)	**84.78±0.43**	**77.08±0.32**	**70.15±0.89**
Polblogs	GCN	94.40±1.47	71.41±2.42	69.16±1.86
	GCN+WGTL$_\text{P}$(ours)	**95.95±0.15**	**73.02±1.13**	**74.52±0.28**
	GAT	95.28±0.51	75.83±0.90	73.11±1.20
	GAT+WGTL$_\text{P}$(ours)	**95.87±0.26**	**76.05±0.79**	**74.21±0.74**
	GraphSAGE	94.54±0.27	77.44±1.71	74.66±0.85
	GraphSAGE+WGTL$_\text{P}$(ours)	**95.58±0.50**	***78.65±1.32**	***74.93±0.81**
	GCN+GNNGuard	95.03±0.25	73.25 ±0.16	72.76±0.75
	GCN+GNNGuard+WGTL$_\text{P}$(ours)	***96.22±0.25**	**73.62±0.22**	**73.72±1.00**

Table 4: Performance on OGBN-arXiv under PRBCD attack.

Models	Perturbation Rate	
	0%	10%
GCN	27.33	21.56
GCN+WGTL$_\text{P}$ (ours)	**28.32**	**22.89**

Table 5: Efficiency of WGTL. All the times are in seconds.

Datasets/ (# Landmarks)	Landmark selection time	Local feat. comput. time	Global feat. comput. time
Cora-ML/124	0.01±0.01	0.12±0.03	5.11±0.13
Citeseer/105	0.01±0.01	0.16±0.02	5.23±1.22
Polblogs/61	0.01±0.00	0.07±0.01	4.64±0.2
Snap-patents/91	0.03±0.02	0.64±0.00	7.54±1.15
Pubmed/394	0.07±0.01	0.51±0.03	27.83±0.47
OGBN-arXiv/84	1.02 ±0.00	12.79±0.31	83.04±2.19

Mettack and Nettack, do not scale to such large-scale graphs (Geisler et al. 2021). Following Geisler et al. (2021), we train a 3-layer GCN to generate attacks. We then present the comparison between GCN and GCN + WGTL$_\text{P}$ on non-poisoned (0%) and poisoned (10%) perturbed graphs in Table 4. We observe that the GCN equipped with our WGTL outperforms GCN on both clean and perturbed OGBN-arXiv.

Q5. Computational Complexity and Efficiency of WGTL. Landmark selection (top-$|\mathfrak{L}|$ degree nodes) has complexity $\mathcal{O}(N\log(N))$. To compute witness features, we compute (1) landmarks-to-witness distances costing $\mathcal{O}(|\mathfrak{L}|(N+|\mathcal{E}|))$ due to BFS-traversal from landmarks, (2) landmark-to-landmark distances costing $\mathcal{O}(|\mathfrak{L}|^2)$, and finally (3) persistent homology via boundary matrix construction and reduction (Edelsbrunner, Letscher, and Zomorodian 2002). Matrix reduction algorithm costs $\mathcal{O}(\zeta^3)$, where ζ is the #simplices in a filtration. Overall computational complexity of computing witness topological feature on a graph is $\mathcal{O}(|\mathfrak{L}|(N+|\mathcal{E}|)+|\mathfrak{L}|^2+\zeta^3)$.

Table 5 shows the total CPU-time to compute Witness topological features broken down into the time spent to select landmarks, to compute local and global topological features. We find that on all the graphs except Pubmed and OGBN-arXiv, the total computation time is <9 seconds. On Pubmed, it takes $\sim$28 seconds, and on large-scale graph OGBN-arXiv, it takes $\sim$96 seconds. These run times are practical given the scale of these graphs.

5 Conclusion and Future Works

By harnessing the strengths of witness complex to efficiently learn topological representations based on the subset of the most essential nodes as skeleton, we have proposed the novel topological defense against adversarial attacks on graphs, WGTL. WGTL is versatile and can be readily integrated with any GNN architecture or another non-topological defense, leading to substantial gains in robustness. We have derived theoretical properties of WGTL, both at the local and global levels, and have illustrated its utility across a wide range of adversarial attacks.

In future, we plan to explore the utility of WGTL with respect to adversarial learning of time-evolving graphs and hypergraphs. Another interesting research direction is to investigate the linkage between the attacker's budget, number of landmarks, and topological attacks targeting the skeleton shape, that is, topological properties of the graph induced by the most important nodes (landmarks).

References

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.

Arafat, N. A.; Basu, D.; and Bressan, S. 2020. ϵ-net Induced Lazy Witness Complexes on Graphs. *arXiv preprint arXiv:2009.13071*.

Arafat, N. A.; Basu, D.; Gel, Y.; and Chen, Y. 2024. When Witnesses Defend: A Witness Graph Topological Layer for Adversarial Graph Learning.

Benson, A. R.; Abebe, R.; Schaub, M. T.; Jadbabaie, A.; and Kleinberg, J. 2018. Simplicial closure and higher-order link prediction. *Proceedings of the National Academy of Sciences*, 115(48): E11221–E11230.

Borgatti, S. P.; Carley, K. M.; and Krackhardt, D. 2006. On the robustness of centrality measures under conditions of imperfect data. *Social networks*, 28(2): 124–136.

Carriere, M.; Chazal, F.; Glisse, M.; Ike, Y.; Kannan, H.; and Umeda, Y. 2021. Optimizing persistent homology based functions. In *ICML*, 1294–1303.

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.

Chazal, F.; Cohen-Steiner, D.; Guibas, L. J.; and Oudot, S. 2008. The stability of persistence diagrams revisited.

Chen, D.; O'Bray, L.; and Borgwardt, K. 2022. Structure-aware transformer for graph representation learning. In *International Conference on Machine Learning*, 3469–3489. PMLR.

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

Chen, Y.; and Gel, Y. R. 2023. Topological Pooling on Graphs. In *AAAI*, volume 37.

Cohen-Steiner, D.; Edelsbrunner, H.; and Harer, J. 2005. Stability of persistence diagrams. In *Proceedings of the twenty-first annual symposium on Computational geometry*, 263–271.

De Silva, V.; and Carlsson, G. 2004. Topological estimation using witness complexes. In *Proceedings of the First Eurographics conference on Point-Based Graphics*, 157–166. Eurographics Association.

Edelsbrunner; Letscher; and Zomorodian. 2002. Topological persistence and simplification. *Discrete & Computational Geometry*, 28: 511–533.

Engstrom, L.; Ilyas, A.; Santurkar, S.; Tsipras, D.; Tran, B.; and Madry, A. 2019. Adversarial robustness as a prior for learned representations. *arXiv preprint arXiv:1906.00945*.

Entezari, N.; Al-Sayouri, S. A.; Darvishzadeh, A.; and Papalexakis, E. E. 2020. All you need is low (rank) defending against adversarial attacks on graphs. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, 169–177.

Feng, B.; Wang, Y.; and Ding, Y. 2021. UAG: Uncertainty-aware attention graph neural network for defending adversarial attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 7404–7412.

Feng, F.; He, X.; Tang, J.; and Chua, T.-S. 2019. Graph adversarial training: Dynamically regularizing based on graph structure. *IEEE Transactions on Knowledge and Data Engineering*, 33(6): 2493–2504.

Gabrielsson, R. B.; Nelson, B. J.; Dwaraknath, A.; and Skraba, P. 2020. A topology layer for machine learning. In *AISTATS*, 1553–1563.

Gebhart, T.; Schrater, P.; and Hylton, A. 2019. Characterizing the shape of activation space in deep neural networks. In *The 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, 1537–1542.

Geisler, S.; Schmidt, T.; Şirin, H.; Zügner, D.; Bojchevski, A.; and Günnemann, S. 2021. Robustness of graph neural networks at scale. *Advances in Neural Information Processing Systems*, 34: 7637–7649.

Goibert, M.; Ricatte, T.; and Dohmatob, E. 2022. An Adversarial Robustness Perspective on the Topology of Neural Networks. In *ML Safety Workshop at NeurIPS 2022*.

Günnemann, S. 2022. Graph neural networks: Adversarial robustness. *Graph Neural Networks: Foundations, Frontiers, and Applications*, 149–176.

Hajij, M.; Zamzmi, G.; Papamarkou, T.; Miolane, N.; Guzmán-Sáenz, A.; Ramamurthy, K. N.; Birdal, T.; Dey, T. K.; Mukherjee, S.; Samaga, S. N.; Livesay, N.; Walters, R.; Rosen, P.; and Schaub, M. 2023. Topological Deep Learning: Going Beyond Graph Data. *arXiv:2206.00606v3*.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

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

Horn, M.; De Brouwer, E.; Moor, M.; Moreau, Y.; Rieck, B.; and Borgwardt, K. 2022. Topological Graph Neural Networks. In *International Conference on Learning Representations*.

Hu, X.; Li, F.; Samaras, D.; and Chen, C. 2019. Topology-preserving deep image segmentation. In *NeurIPS*, volume 32.

In, Y.; Yoon, K.; Kim, K.; Shin, K.; and Park, C. 2024. Self-Guided Robust Graph Structure Refinement. In *Proceedings of the ACM on Web Conference 2024*, 697–708.

Jia, Y.; Zou, D.; Wang, H.; and Jin, H. 2023. Enhancing Node-Level Adversarial Defenses by Lipschitz Regularization of Graph Neural Networks. In *SIGKDD*, 951–963.

Jin, W.; Derr, T.; Wang, Y.; Ma, Y.; Liu, Z.; and Tang, J. 2021a. Node similarity preserving graph convolutional networks. In *Proceedings of the 14th ACM international conference on web search and data mining*, 148–156.

Jin, W.; Li, Y.; Xu, H.; Wang, Y.; Ji, S.; Aggarwal, C.; and Tang, J. 2021b. Adversarial attacks and defenses on graphs. *ACM SIGKDD Explorations Newsletter*, 22(2): 19–34.

Jin, W.; Ma, Y.; Liu, X.; Tang, X.; Wang, S.; and Tang, J. 2020. Graph structure learning for robust graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 66–74.

Kipf, T. N.; and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Kong, K.; Li, G.; Ding, M.; Wu, Z.; Zhu, C.; Ghanem, B.; Taylor, G.; and Goldstein, T. 2020. Flag: Adversarial data augmentation for graph neural networks.

Liu, A.; Li, W.; Li, T.; Li, B.; Huang, H.; and Zhou, P. 2023. Towards Inductive Robustness: Distilling and Fostering Wave-induced Resonance in Transductive GCNs Against Graph Adversarial Attacks. *arXiv preprint arXiv:2312.08651*.

Mujkanovic, F.; Geisler, S.; Günnemann, S.; and Bojchevski, A. 2022. Are Defenses for Graph Neural Networks Robust? *Advances in Neural Information Processing Systems*, 35: 8954–8968.

Rieck, B.; Yates, T.; Bock, C.; Borgwardt, K.; Wolf, G.; Turk-Browne, N.; and Krishnaswamy, S. 2020. Uncovering the topology of time-varying fMRI data using cubical persistence. *Advances in neural information processing systems*, 33: 6900–6912.

Sun, L.; Dou, Y.; Yang, C.; Zhang, K.; Wang, J.; Philip, S. Y.; He, L.; and Li, B. 2022. Adversarial attack and defense on graph data: A survey. *IEEE Transactions on Knowledge and Data Engineering*.

Tang, X.; Li, Y.; Sun, Y.; Yao, H.; Mitra, P.; and Wang, S. 2020. Transferring robustness for graph neural network against poisoning attacks. In *Proceedings of the 13th international conference on web search and data mining*, 600–608.

Torres, L.; Blevins, A. S.; Bassett, D.; and Eliassi-Rad, T. 2021. The why, how, and when of representations for complex systems. *SIAM Review*, 63(3): 435–485.

Wu, F.; Souza, A.; Zhang, T.; Fifty, C.; Yu, T.; and Weinberger, K. 2019a. Simplifying graph convolutional networks. In *International conference on machine learning*, 6861–6871. PMLR.

Wu, H.; Wang, C.; Tyshetskiy, Y.; Docherty, A.; Lu, K.; and Zhu, L. 2019b. Adversarial examples for graph data: deep insights into attack and defense. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 4816–4823.

Xu, K.; Chen, H.; Liu, S.; Chen, P.-Y.; Weng, T.-W.; Hong, M.; and Lin, X. 2019. Topology Attack and Defense for Graph Neural Networks: An Optimization Perspective. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

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 extended persistent homology on graphs. In *ICLR 2022 Workshop on Geometrical and Topological Representation Learning*.

Zhang, A.; and Ma, J. 2020. Defensevgae: Defending against adversarial attacks on graph data via a variational graph autoencoder. *arXiv preprint arXiv:2006.08900*.

Zhang, X.; and Zitnik, M. 2020. Gnnguard: Defending graph neural networks against adversarial attacks. *Advances in neural information processing systems*, 33: 9263–9275.

Zhao, Q.; and Wang, Y. 2019. Learning metrics for persistence-based summaries and applications for graph classification. *Advances in Neural Information Processing Systems*, 32.

Zhu, D.; Zhang, Z.; Cui, P.; and Zhu, W. 2019. Robust graph convolutional networks against adversarial attacks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 1399–1407.

Zomorodian, A. 2010. Fast construction of the Vietoris-Rips complex. *Computers & Graphics*, 34(3): 263–271.

Zomorodian, A. J. 2005. *Topology for computing*, volume 16. Cambridge university press.

Zügner, D.; Akbarnejad, A.; and Günnemann, S. 2018. Adversarial attacks on neural networks for graph data. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2847–2856.

Zügner, D.; and Günnemann, S. 2019a. Adversarial Attacks on Graph Neural Networks via Meta Learning. In *International Conference on Learning Representations (ICLR)*.

Zügner, D.; and Günnemann, S. 2019b. Certifiable robustness and robust training for graph convolutional networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 246–256.