GRIL: A 2-parameter Persistence Based Vectorization for Machine Learning

Cheng Xin [*1] **Soham Mukherjee** [*1] **Shreyas N. Samaga** [1] **Tamal K. Dey** [1]

Abstract

1-parameter persistent homology, a cornerstone in Topological Data Analysis (TDA), studies the evolution of topological features such as connected components and cycles hidden in data. It has been applied to enhance the representation power of deep learning models, such as Graph Neural Networks (GNNs). To enrich the representations of topological features, here we propose to study 2-parameter persistence modules induced by bi-filtration functions. In order to incorporate these representations into machine learning models, we introduce a novel vector representation called Generalized Rank Invariant Landscape (GRIL) for 2-parameter persistence modules. We show that this vector representation is 1-Lipschitz stable and differentiable with respect to underlying filtration functions and can be easily integrated into machine learning models to augment encoding topological features. We present an algorithm to compute the vector representation efficiently. We also test our methods on synthetic and benchmark graph datasets, and compare the results with previous vector representations of 1-parameter and 2-parameter persistence modules. Further, we augment GNNs with GRIL features and observe an increase in performance indicating that GRIL can capture additional features enriching GNNs. We make the complete code for the proposed method available at https://github.com/soham0209/mpml-graph.

1. Introduction

Machine learning models such as Graph Neural Networks (GNNs) (Gori et al., 2005; Scarselli et al., 2009; Kipf & Welling, 2017; Xu et al., 2019) are well-known successful tools from the geometric deep learning community. Some recent research has indicated that the representation power of such models can be augmented by infusing topological information (Hofer et al., 2017; Dehmamy et al., 2019; Carrière et al., 2020; Horn et al., 2022). One way to do that is by applying *persistent homology*, which is a powerful tool for characterizing the shape of data, rooted in the theory of algebraic topology. It has spawned the flourishing area of Topological Data Analysis. The classical persistent homology, also known as, 1-parameter *persistence module*, has attracted plenty of attention from both theory and applications (Edelsbrunner & Harer, 2010; Oudot, 2015; Carlsson & Vejdemo-Johansson, 2021; Dey & Wang, 2022). In essence, a 1-parameter persistence homology captures the evolution of some topological information within a topological space $\mathcal{X}$ along an ascending filtration determined by a scalar function $\mathcal{X} \to \mathbb{R}$. It can be losslessly summarized by a complete discrete invariant such as a *persistence diagram*, *rank invariant* or *barcode*. In recent years, many works have successfully integrated persistence homology with machine learning models (Corbet et al., 2019; Chen et al., 2019; Carrière et al., 2020; Kim et al., 2020; Gabrielsson et al., 2020; Zhao et al., 2020; Hofer et al., 2020; Swenson et al., 2020; Carrière & Blumberg, 2020; Vipond, 2020; Bouritsas et al., 2022; Horn et al., 2022; Cang & Wei, 2017; Demir et al., 2022; Zhang et al., 2022; Liu et al., 2022).

To further enhance the capacity of persistent homology, it is natural to consider a more general multivariate filtration function $\mathcal{X} \to \mathbb{R}^d$ for $d \geq 2$ in place of a real valued function, and represent its topological information by multiparameter persistence modules. However, the structure of multiparameter persistence modules is much more complicated than 1-parameter persistence modules. In 1-parameter case, the modules are completely characterized by what is called *barcode* or *persistence diagram* (Chazal et al., 2009a; Lesnick, 2015). Unfortunately, there is no such discrete complete invariant which can summarize multiparameter persistence modules completely (Carlsson & Zomorodian, 2009). Given this limitation, building a useful vector representation from multiparameter persistence modules while capturing as much topological information as possible for machine learning models becomes an important but challenging problem.

To address this challenge, different kinds of vector repre-

[*]Equal contribution [1]Department of Computer Science, Purdue University, West Lafayette, USA. Correspondence to: <{xinc, mukher26, ssamaga, tamaldey}@purdue.edu>.

Proceedings of the 2^{nd} Annual Workshop on Topology, Algebra, and Geometry in Machine Learning (TAG-ML) at the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. 2023.

sentations have been proposed for 2-parameter persistence modules (Corbet et al., 2019; Vipond, 2020; Carrière & Blumberg, 2020). All these works are essentially based on the invariant called fibered (sliced) barcodes (Lesnick & Wright, 2015). However, such representations capture as much topological information as determined by the well-known incomplete summary called rank invariant (Carlsson & Zomorodian, 2009) which is equivalent to fibered barcodes.

In this paper, we propose a new vector representation to extend its expressive power in terms of capturing topological information from a 2-parameter persistence module:

- We introduce *Generalized Rank Invariant Landscape* (GRIL), a new vector representation encoding richer information beyond fibered barcodes for 2-parameter persistence modules, based on the idea of *generalized rank invariant* (Kim & Mémoli, 2021) and its computation by zigzag persistence (Dey et al., 2022). The construction of GRIL can be viewed as a generalization of persistence landscape (Bubenik, 2015; Vipond, 2020), hence has more discriminating power.
- We show that this vector representation GRIL is 1-Lipschitz stable and differentiable with respect to the filtration function f, which allows one to build a topological representation as a machine learning model.
- We propose an efficient algorithm to compute (GRIL), demonstrate its use on synthetic and benchmark graph datasets, and compare the results with previous vector representations of 1-parameter and 2-parameter persistence modules. Specifically, we present results indicating that GNNs may improve when augmented with GRIL features for graph classification task.

2. Background

In this section, we start with an overview of single and multiparameter persistence modules followed by formal definitions of basic concepts. Then we provide a high-level idea of how to construct our vector representation GRIL. For a more comprehensive introduction to persistence modules, we refer the interested reader to (Edelsbrunner & Harer, 2010; Oudot, 2015; Carlsson & Vejdemo-Johansson, 2021; Dey & Wang, 2022).

The standard pipeline of 1-parameter persistence module is as follows: Given a domain of interest $\mathcal{X}$ (e.g. a topological space, point cloud data, a graph, or a simplicial complex) with a scalar function $f : \mathcal{X} \to \mathbb{R}$, one filters the domain $\mathcal{X}$ by the sublevel sets $\mathcal{X}_\alpha \triangleq \{x \in \mathcal{X} \mid f(x) \leq \alpha\}$ along with a continuously increasing threshold $\alpha \in \mathbb{R}$. The collection $\{\mathcal{X}_\alpha\}$, which is called a *filtration*, forms an increasing sequence of subspaces $\emptyset = \mathcal{X}_{-\infty} \subseteq \mathcal{X}_{\alpha_1} \subseteq \cdots \subseteq \mathcal{X}_{+\infty} = \mathcal{X}$. Along with the filtration, topological features appear, persist, and disappear over a collection of intervals. We consider pth homology groups $H_p(-)$ over a field, say $\mathbb{Z}_2$, of the subspaces in this filtration, which results into a sequence of vector spaces. These vector spaces are connected by inclusion-induced linear maps forming an algebraic structure $0 = H_p(\mathcal{X}_{-\infty}) \to H_p(\mathcal{X}_{\alpha_1}) \to \cdots \to H_p(\mathcal{X}_{+\infty})$. (see (Hatcher, 2000)). This algebraic structure, known as 1-parameter persistence module induced by f and denoted as M^f, can be uniquely decomposed into a collection of atomic modules called interval modules, which completely characterizes the topological features in regard to the three behaviors–appearance, persistence, and disappearance of all p-dimensional cycles. This unique decomposition of a 1-parameter persistence module is commonly summarized as a complete discrete invariant, *persistence diagram* (Edelsbrunner et al., 2000) or *barcode* (Zomorodian & Carlsson, 2005). Figure 1 (left) shows a filtration of a simplicial complex that induces a 1-parameter persistence module and its decomposition into bars.

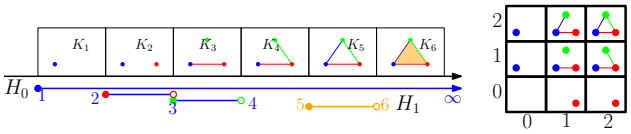

Figure 1: (left) 1-parameter filtration and bars; (right) a 2-parameter filtration inducing a 2-parameter persistence module whose decomposition is not shown.

Some problems in practice may demand tracking the topological information in a filtration that is not necessarily linear. For example, in (Adcock et al., 2014), 2-parameter persistence modules are shown to be better for classifying hepatic lesions compared to 1-parameter persistence modules. In (Keller et al., 2018; Demir et al., 2022), a virtual screening system based on 2-parameter persistence modules are shown to be effective for searching new candidate drugs. In such applications, instead of studying a sequential filtration filtered by a scalar function, one may study a grid-filtration induced by a $\mathbb{R}^2$-valued bi-filtration function $f : \mathcal{X} \to \mathbb{R}^2$ with $\mathbb{R}^2$ equipped with partial order $\boldsymbol{u} \leq \boldsymbol{v} : u_1 \leq v_1, u_2 \leq v_2$; see Figure 1(right) for an example of 2-parameter filtration. Following a similar pipeline as the 1-parameter persistence module, one will get a collection of vector spaces $\{M^f_{\boldsymbol{u}}\}_{\boldsymbol{u}\in\mathbb{R}^2}$ indexed by vectors $\boldsymbol{u} = (u_1, u_2) \in \mathbb{R}^2$ and linear maps $\{M^f_{\boldsymbol{u}\to\boldsymbol{v}} : M^f_{\boldsymbol{u}} \to M^f_{\boldsymbol{v}} \mid \boldsymbol{u} \leq \boldsymbol{v} \in \mathbb{R}^2\}$ for all comparable $\boldsymbol{u} \leq \boldsymbol{v}$. The entire structure M^f, in analogy to the 1-parameter case, is called a 2-parameter persistence module induced from f. Unlike 1-parameter case, there is no *complete* discrete invariant like persistence diagrams or barcodes that can losslessly summarize the whole structure of 2-parameter persistence modules (Carlsson & Zomorodian,

2009). A good non-complete invariant for 2-parameter persistence modules should characterize many non-isomorphic topological features, ideally as many as possible. At the same time, it should be stable with respect to small perturbations of filtration functions, which guarantees its important properties of continuity and differentiability for machine learning models. Therefore, building a good summary in general for 2-parameter persistence modules which is also applicable to machine learning models is an important and challenging problem.

Overview: Our approach computes a *landscape function* over the 2-parameter domain and then vectorizes it. At this high level, this is similar to the approach in (Vipond, 2020). However, the landscape function we construct is much more general and thus potentially has the power of capturing more topological information. In particular, we use the concept of *generalized rank invariant* introduced in (Kim & Mémoli, 2021), which indeed generalizes the traditional rank invariant used in (Vipond, 2020). As opposed to simple rank invariant which is defined over rectangles, generalized ranks are defined over their generalizations called *intervals*. We define it more formally in section 3 below.

One difficulty facing the use of the generalized ranks in TDA was that its efficient computation was not known. Recently, in (Dey et al., 2022), the authors showed that generalized ranks for intervals in 2-parameter persistence modules can be obtained by considering a persistence module supported on a linear poset induced by the boundary of the interval in question. However, this linear poset is not totally ordered as in 1-parameter persistence, and thus gives rise to what is called *zigzag persistence* (Carlsson & De Silva, 2010) where the inclusions can both be in forward and backward directions unlike traditional 1-parameter persistence where they are only in forward directions; With this result, computing generalized ranks efficiently boils down to computing zigzag persistence efficiently. For this purpose, we use a recently discovered fast zigzag algorithm and its efficient implementation (Dey & Hou, 2022)[1].

Our method samples a subset of grid points from the 2-parameter grid spanned by a given bi-filtration function, and computes the landscape function values (Definition 3.2) at those points based on generalized ranks. For this, the algorithm considers an expanding sequence of intervals which we call *worms* centered at each point $\boldsymbol{p}$ and computes generalized rank over them to determine the 'width' of the maximal worm sustaining a chosen rank. This maximization is achieved by a binary search over the sequence of worms centering $\boldsymbol{p}$; section 4 describes this procedure. The widths, thus computed for each sample point, constitute the landscape function values which become the basis for our vector representation.

[1] https://github.com/taohou01/fzz

3. Generalized Rank Invariant Landscape

In this section, we introduce Generalized Rank Invariant Landscape, abbreviated as GRIL, a stable and differentiable vector representation of 2-parameter persistence modules.

Let $M = M^f$ be a 2-parameter persistence module induced by a filtration function f. The restriction of M to an interval I, denoted as $M|_I$, is the collection of vector spaces $\{M_{\boldsymbol{u}} \mid \boldsymbol{u} \in I\}$ along with linear maps $\{M_{\boldsymbol{u}\to\boldsymbol{v}} \mid \boldsymbol{u} \le \boldsymbol{v} \in I)\}$. One can define the generalized rank of $M|_I$ (Kim & Mémoli, 2021) as the rank of the canonical linear map from limit $\varprojlim M|_I$ to colimit $\varinjlim M|_I$ of $M|_I$ (see Appendix A):

$$\mathsf{rk}^M(I) \triangleq \operatorname{rank}[\varprojlim M|_I \to \varinjlim M|_I]$$

A formal explanation of limit and colimit is beyond the scope of this article; we refer readers to (MacLane, 1971) for their definitions and also the construction of the canonical limit-to-colimit map in category theory. Intuitively, $\mathsf{rk}^M(I)$ captures the number of *independent* topological features encoded in M with the support over the entire interval I. Specially, when $I = [\boldsymbol{u}, \boldsymbol{v}] \triangleq \{\boldsymbol{w} \in \mathbb{R}^2 \mid \boldsymbol{u} \le \boldsymbol{w} \le \boldsymbol{v}\}$ is a rectangle, $\varprojlim M|_I = M_{\boldsymbol{u}}$ and $\varinjlim M|_I = M_{\boldsymbol{v}}$. Then $\mathsf{rk}^M(I)$ equals the traditional rank of the linear map $M_{\boldsymbol{u}\to\boldsymbol{v}}$.

Remark 3.1. An interesting property of the generalized rank invariant is that its value over a larger interval is less than or equal to its value over any interval contained inside the larger interval. Formally, $I \subseteq J \implies \mathsf{rk}^M(I) \ge \mathsf{rk}^M(J)$. We implicitly use this *monotone* property in the definition of GRIL.

The basic idea of GRIL is to consider a collection of generalized ranks $\{\mathsf{rk}^M(I)\}_{I\in\mathcal{W}}$ over some covering set $\mathcal{W}$ on $\mathbb{R}^2$, which is called a *generalized rank invariant* of M over $\mathcal{W}$. Let $\boxed{\boldsymbol{p}}_\delta \triangleq \{\boldsymbol{w} : \|\boldsymbol{p} - \boldsymbol{w}\|_\infty \le \delta\}$ be the δ-square centered at $\boldsymbol{p}$ with side 2δ. For given $\boldsymbol{p} \in \mathbb{R}^2, \ell \ge 1, \delta > 0$, we define an *$\ell$-worm* $\boxed{\boldsymbol{p}}^\ell_\delta$ to be the union over all δ-squares $\boxed{\boldsymbol{q}}_\delta$ centered at some point $\boldsymbol{q}$ on the off-diagonal line segment $\boldsymbol{p} + \alpha \cdot (1, -1)$ with $|\alpha| \le (\ell - 1)\delta$. See Figure 2 for an illustration.

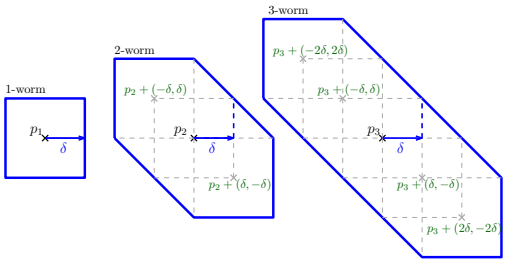

Figure 2: Examples of three ℓ-worms with $\ell = 1, 2, 3$.

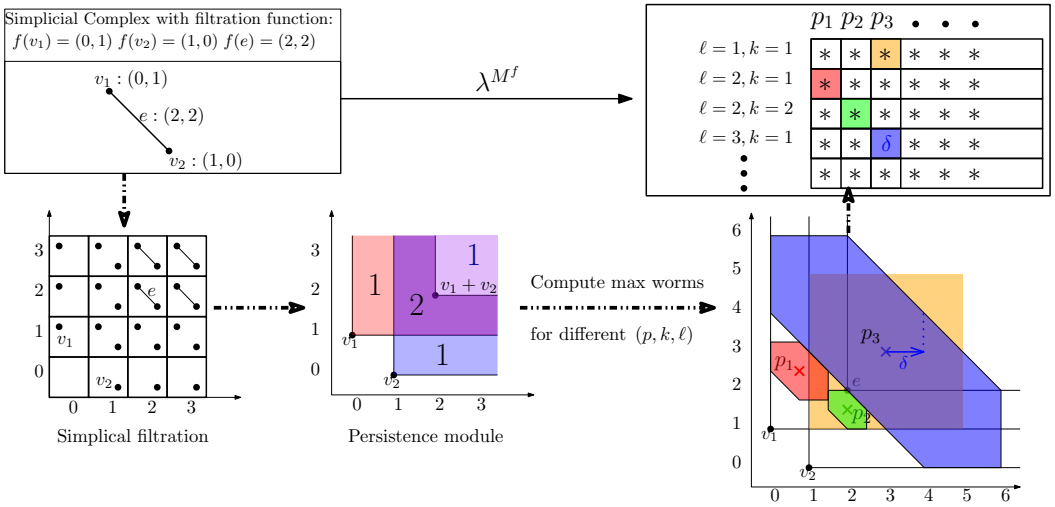

Figure 3: The construction starts from a simplicial complex with a bi-filtration function as shown on the top left. The simplicial complex consists of two vertices connected by one edge. Based on the bi-filtration, a simplicial bi-filtration can be defined as shown on the bottom left. On the mid bottom, a 2-parameter persistence module is induced from the above simplicial filtration. If we check the dimensions of the vector spaces on all points of the plane, there are 1-dimensional vector spaces on red, blue and light purple regions. On the L-shaped dark purple region, the vector spaces have dimension 2. For this 2-parameter persistence module, we calculate $\lambda^{M^f}(\boldsymbol{p}, k, \ell)$ for all tuples $(\boldsymbol{p}, k, \ell) \in \mathcal{P} \times K \times L$ to get our GRIL vector representation. By Definition 3.2 the value $\lambda^{M^f}(\boldsymbol{p}, k, \ell)$ corresponds to the width of the maximal ℓ-worm on which the generalized rank is at least k. On the bottom right, the interval in red is the maximal 2-worm for $\lambda^{M^f}(\boldsymbol{p}_1, k = 1, \ell = 2)$. The green interval is the maximal 2-worm for $\lambda^{M^f}(\boldsymbol{p}_2, k = 2, \ell = 2)$. The yellow square is the maximal 1-worm for $\lambda^{M^f}(\boldsymbol{p}_3, k = 1, \ell = 1)$, and the blue interval is the maximal 3-worm for $\lambda^{M^f}(\boldsymbol{p}_3, k = 1, \ell = 3)$. Finally, on the top right, we have our GRIL vector representation λ^{M^f} which is a collection of vectors. Each vector corresponding to a different ℓ and k consists of values as the width of maximal worms at each center point $\boldsymbol{p}$. As an example, the blue one on the last vector at position p_3 has value δ which is the width of the blue worm.

Formally,

$$\boxed{\boldsymbol{p}}_\delta^\ell \triangleq \bigcup_{\substack{\boldsymbol{q}=\boldsymbol{p}+(\alpha,-\alpha) \\ |\alpha| \leq (l-1)\delta}} \boxed{\boldsymbol{q}}_\delta$$

We call $\boldsymbol{p}$ the *center point* and δ the *width* of the ℓ-worm $\boxed{\boldsymbol{p}}_\delta^\ell$. As a special case, when $\ell = 1$, $\boxed{\boldsymbol{p}}_\delta^1 = \boxed{\boldsymbol{p}}_\delta$ is just the δ-square with side 2δ.

We choose $\mathcal{W}$ to be a set of *Worms* defined as follows:

$$\mathcal{W} \triangleq \left\{ W = \boxed{\boldsymbol{p}}_\delta^\ell \mid \delta > 0, \ell \geq 1, \boldsymbol{p} \in \mathbb{R}^2 \right\}$$

Now we are ready to define the main construct in this paper which uses the monotone property of generalized rank mentioned in Remark 3.1.

Definition 3.2 (Generalized Rank Invariant Landscape (GRIL))**.** For a persistence module M, the *Generalized Rank Invariant Landscape (*GRIL*)* of M is a function $\lambda^M : \mathbb{R}^2 \times \mathbb{N}_+ \times \mathbb{N}_+ \to \mathbb{R}$ defined as

$$\lambda^M(\boldsymbol{p}, k, \ell) \triangleq \sup_{\delta \geq 0} \{ \mathsf{rk}^M(\boxed{\boldsymbol{p}}_\delta^\ell) \geq k \}. \tag{1}$$

We can see from the definition that given a persistence module M, a point $\boldsymbol{p}$, a rank k and ℓ, the value of GRIL $(\lambda^M(\boldsymbol{p}, k, \ell))$ is, in essence, the width δ of the "maximal" ℓ-worm $W = \boxed{\boldsymbol{p}}_\delta^\ell$ centered at $\boldsymbol{p}$ such that the value of the generalized rank over W is greater than or equal to k. See Figure 3 bottom right for some examples of maximal worms.

It turns out that, GRIL as an invariant is equivalent to the generalized rank invariant over $\mathcal{W}$.

Proposition 3.3. GRIL *is equivalent to the generalized rank invariant over* $\mathcal{W}$*. Here the equivalence means bijective reconstruction from each other.*

See Figure 3 for an illustration of the overall pipeline of our construction of λ^M starting from a filtration function on a simplicial complex. Figure 4 shows the discriminating power of GRIL where we see that GRIL can differentiate between shapes that are topologically non-equivalent.

Stability of GRIL. An important property of GRIL is its *stability property* which makes it immune to small perturba-

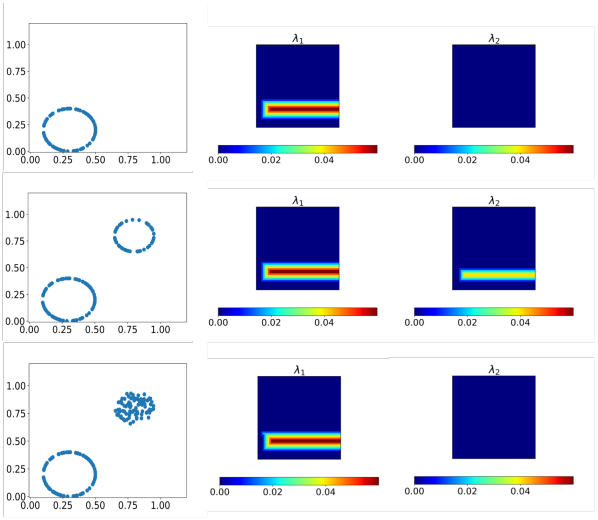

Figure 4: GRIL as a topological discriminator: each row shows a point cloud, GRIL value heatmap for ranks $k = 1$ and $k = 2$ in homology of degree 1 denoted as λ_1 and λ_2 respectively; (row 1) first Betti number (β_1) of a circle is 1 which is reflected in λ_1 being non-zero; (row 2) β_1 for two circles is 2 which is reflected in both λ_1 and λ_2 being non-zero; (row 3) β_1 of a circle and disk together is 1 which is reflected in λ_1 being non-zero but λ_2 being zero.

tions of the input bi-filtration while still retaining the ability to characterize topologies. We will show GRIL is stable with respect to input filtrations.

Proposition 3.4 (Stability)**.** *Given two filtration functions* $f, f' : \mathcal{X} \to \mathbb{R}^2$,

$$\left\| \lambda^{M^f} - \lambda^{M^{f'}} \right\|_\infty \leq \|f - f'\|_\infty$$

Proposition 3.5 (Lipschitz continuous)**.** *For a finite space* $\mathcal{X}$ *with* $|\mathcal{X}| = n$ *and fixed* $k, \ell, \boldsymbol{p}$*, the function* $\Lambda_{\boldsymbol{p}}^{k,\ell} : \mathbb{R}^{2n} \to \mathbb{R}$ *given by* $\Lambda_{\boldsymbol{p}}^{k,\ell}(f) = \lambda^{M^f}(k, \ell, \boldsymbol{p})$ *is Lipschitz continuous.*

Proof. Given filtration functions f, f' and their corresponding vector representations $v_f, v_{f'} \in \mathbb{R}^{2n}$, one can check that $\|f - f'\|_\infty \leq 2\|v_f - v_{f'}\|_\infty \leq 2\|v_f - v_{f'}\|_2$. Combining this with Proposition 3.4, we get that $\Lambda_{\boldsymbol{p}}^{k,\ell}$ is Lipschitz continuous with respect to the underlying filtration functions. □

Corollary 3.6. $\Lambda_{\boldsymbol{p}}^{k,\ell}$ *is differentiable almost everywhere.*

By Rademacher's theorem (Evans & Gariepy, 2015), we have $\Lambda_{\boldsymbol{p}}^{k,\ell}$, as a Lipschitz continuous function, being differentiable almost everywhere. More discussions on the stability and differentiability of GRIL are in Appendix B and E.

4. Algorithm

We present our algorithm to compute GRIL in this section.

In practice, we choose center points $\boldsymbol{p}$ from some finite subset $\mathcal{P} \subset \mathbb{R}^2$, e.g. a finite uniform grid in $\mathbb{R}^2$, and consider $k \leq K, \ell \leq L$ for some fixed $K, L \in \mathbb{N}_+$. Then, GRIL $\{\lambda^M(\boldsymbol{p}, k, \ell)\}$ can be viewed as a vector of dimension $|\mathcal{P}| \times K \times L$.

The high-level idea of the algorithm is as follows: Given a bi-filtration function $f : \mathcal{X} \to \mathbb{R}^2$, for each triple $(\boldsymbol{p}, k, \ell) \in \mathcal{P} \times K \times L$, we compute $\lambda^{M^f}(\boldsymbol{p}, k, \ell) = \sup_{\delta \geq 0}\{\mathsf{rk}^{M^f}(\boxed{\boldsymbol{p}}_\delta^\ell) \geq k\}$. In essence, we need to compute the maximum width over worms on which the generalized rank is at least k. In order to find the value of this width, we use binary search. We compute generalized rank $\mathsf{rk}^{M^f}\left(\boxed{\boldsymbol{p}}_\delta^\ell\right)$ by applying the algorithm proposed in (Dey et al., 2022), which uses zigzag persistence on a boundary path. This zigzag persistence is computed efficiently by a recent algorithm proposed in (Dey & Hou, 2022). We denote the sub-routine to compute generalized rank over a worm by COMPUTERANK in algorithm 1 mentioned below. COMPUTERANK(f, I) takes as input a bi-filtration function f and an interval I, and outputs generalized rank over that interval. In order to use the algorithm proposed in (Dey et al., 2022), the worms need to have their boundaries aligned with a grid structure defined on the range of f. Thus, we normalize f to be in the range $[0, 1] \times [0, 1]$, define a grid structure on $[0, 1] \times [0, 1]$ and discretize the worms. Let GRID $= \{\left(\frac{m}{M}, \frac{n}{M}\right) \mid m, n \in \{0, 1, \ldots, M\}\}$ for some $M \in \mathbb{Z}_+$. We denote the grid resolution as $\rho \triangleq 1/M$. We take the set of center points $\mathcal{P} \subseteq$ GRID as a uniform subgrid of GRID. We consider the discrete worms for $\boldsymbol{p} \in \mathcal{P}, \delta = d \cdot \rho, d \in \mathbb{Z}_{\geq 0}$ as follows:

$$\widehat{\boxed{\boldsymbol{p}}}_\delta^\ell \triangleq \bigcup_{\substack{\boldsymbol{q} = \boldsymbol{p} + (\alpha, -\alpha) \\ |\alpha| \leq (l-1)\delta \\ \boldsymbol{q} \in \text{GRID}}} \boxed{\boldsymbol{q}}_\delta. \tag{2}$$

Essentially, a discrete ℓ-worm $\widehat{\boxed{\boldsymbol{p}}}_\delta^\ell$ centered at $\boldsymbol{p}$ with width δ is the union of $2\ell - 1$ squares with width δ centered at $\boldsymbol{p} \pm (c\delta, -c\delta)$ for $c \in \{0, 1, \ldots, \ell - 1\}$ along with the intermediate staircases between two consecutive squares of step-size equal to *grid resolution* (ρ). Figure 5 (middle) shows the discretization of a 2-worm. This construction is sensitive to the grid resolution.

Now all such discrete worms $\widehat{\boxed{\boldsymbol{p}}}$ are intervals whose boundaries are aligned with the GRID. We apply the procedure COMPUTERANK(f, I) to compute $\mathsf{rk}^{M^f}(I)$ for $I = \widehat{\boxed{\boldsymbol{p}}}_\delta^\ell$. Denote

$$\hat{\lambda}^{M^f}(\boldsymbol{p}, k, \ell) = \sup_{\delta \geq 0}\{\mathsf{rk}^{M^f}(\widehat{\boxed{\boldsymbol{p}}}_\delta^\ell) \geq k\}. \tag{3}$$

Remark 4.1. One can observe that

$$\lambda^{M^f}(\boldsymbol{p}, k, \ell) \leq \hat{\lambda}^{M^f}(\boldsymbol{p}, k, \ell) \leq \lambda^{M^f}(\boldsymbol{p}, k, \ell) + \rho$$

Therefore, we compute $\hat{\lambda}$ as an approximation of λ in practice.

The pseudo-code is given in Algorithm 1. The algorithm is described in detail in Appendix D.

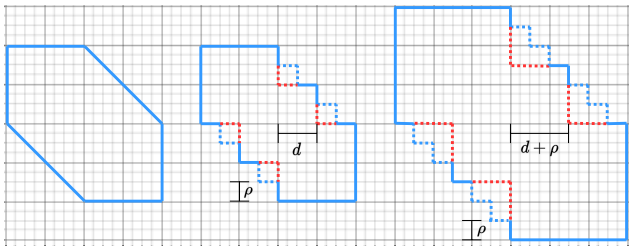

Figure 5: A 2-worm, discretized 2-worm and expanded discretized 2-worm; ρ denotes grid resolution. The blue dotted lines show the intermediate staircase with step-size ρ. The red dotted lines form parts of the squares with size d which are replaced by the blue dotted lines in the worm. The last figure shows the expanded 2-worm with red and blue dotted lines. The expanded 2-worm has width $d + \rho$ which is the one step expansion of the worm with width d.

Algorithm 1 COMPUTEGRIL

Input: f : Bi-filtration function, $\ell \geq 0, k \geq 1, \boldsymbol{p} \in \mathcal{P} \subseteq$ GRID, ρ: grid resolution
Output: $\hat{\lambda}(\boldsymbol{p}, k, l)$: GRIL value at $\boldsymbol{p}$ for fixed k and ℓ
Initialize: $d_{min} \leftarrow \rho, d_{max} \leftarrow 1, \lambda \leftarrow 0$
while $d_{min} \leq d_{max}$ **do**
  $d \leftarrow (d_{min} + d_{max})/2$; $I \leftarrow \widehat{\boxed{\boldsymbol{p}}}^{\ell}_{d}$.
  $r \leftarrow$ COMPUTERANK(f, I)
  if $r \geq k$ **then**
    $\lambda \leftarrow d$; $d_{min} \leftarrow d + \rho$
  else
    $d_{max} \leftarrow d - \rho$
  end if
end while
return λ

Time complexity. Assuming a grid with t nodes and a bi-filtration of a complex with n simplices on it, one can observe that each probe in the binary search takes $O(n^\omega)$ time where $\omega < 2.37286$ is the matrix multiplication exponent (Alman & Williams, 2021). This is because each probe generates a zigzag filtration of length $O(n)$ with $O(n)$ simplices. Therefore, the binary search takes $O(n^\omega \log t)$ time giving a total time complexity of $O(tn^\omega \log t))$ that accounts for $O(t)$ worms.

Speeding up the implementation. In implementation, we use some observations that help run COMPUTEGRIL more efficiently in practice. When computing GRIL for $k = 1, 2, \ldots, n$, we use the monotone property described in Remark 3.1 to reduce the scope of the binary search for successive values of k. For example, the value of GRIL for k is always greater than or equal to the value of GRIL for $k + 1$. Thus, we can reduce the scope of the binary search while computing for $k + 1$ by setting the maximum in the binary search to be the value of GRIL at k. Further, we store the values of rank for a given width d while computing the value of GRIL for a k. This information can be reused in later computations. For example, we store the values of generalized ranks of worms for different values of d at a center point $\boldsymbol{p}$ during the binary search for, say $k = k_0$. We use this information for successive binary searches for all $k > k_0$ and save on the zigzag persistence computation for those values of d. While computing zigzag persistent, along with the barcode for 0th homology group, the barcode for 1st homology group is also computed. We store this information and reuse it while computing GRIL values for 1st homology group. These observations reduce the total number of zigzag persistence computations to a significant extent resulting in reducing the total computational time.

5. Experiments

Our method GRIL exploits generalized rank invariant whereas existing methods exploit rank invariant which is equivalent to fibered barcode. Although both invariants are known to be incomplete for multiparameter persistence as any other discrete invariant, the generalized rank invariant is more informative in theory. Our experiments support this theoretical hypothesis in practice to some extent as we obtain better accuracy for all cases in Table 1 and 13 out of 20 cases in Table 2 in comparison to existing methods applying some form of fibered barcodes. We perform experiments on synthetic datasets as well as graph benchmark datasets. On these datasets, we define a bi-filtration and compute GRIL values $\lambda(\boldsymbol{p}, k, \ell)$ for $\ell = 2$ and for each $k \in \{1, 2, \ldots, 5\}$ where $\boldsymbol{p}$ is chosen over a uniform subgrid. Some datasets require a finer resolution for capturing meaningful information while for others, finer resolutions capture redundant information and a coarser resolution performs better. Therefore, we sample subgrids with different step-sizes from the discretized grid described in section 4 and vary $\boldsymbol{p}$ over these subgrids.We first describe an experiment on a synthetic data set and follow it with experiments on benchmark graph data sets.

5.1. Experiment with HourGlass dataset

We test our model on a synthetic dataset (HourGlass) that entails a binary graph classification problem over a collec-

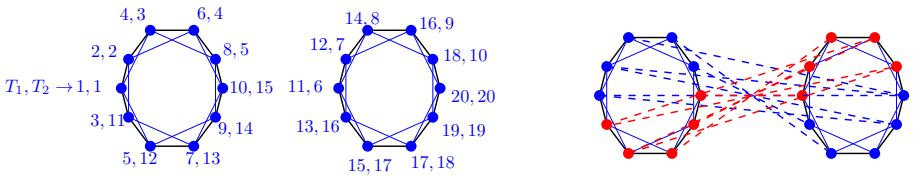

Figure 6: (Left) An example of a graph consisting of two circulant subgraphs. The pair of indices on each node represents the its order on the traversals T_1 and T_2 respectively. Both traversals start from the left node as the root node. (Right) Cross edges placed across two subgraphs.

tion of attributed undirected graphs. Note that this synthetic dataset is designed to show that some attributed graphs can be easily classfied by 2-parameter persistence modules but not so by 1-parameter persistence modules or commonly used GNN models. Each graph G from either class is composed with two circulant subgraphs G_1, G_2 connected by some cross edges. The node attributes are order indices generated by two different traversals T_1, T_2. The label of classes corresponds to these two different traversals T_1, T_2. Therefore, the classification task is that given an attributed graph G, the model needs to predict which traversal is used to generate G. See Figure 6 (left) as an example of two attributed graphs with the same graph structure but with different node attributes generated by two different traversals. More details can be found in Appendix C.1. We denote HourGlass[a,b] as the dataset of graphs generated with node size of each circulant subgraphs in range $[a, b]$. We generate three datasets with different sizes: HourGlass[10,20], HourGlass[21,30], HourGlass[31,40]. Each dataset contains roughly 400 graphs. We evenly split HourGlass[21,30] into balanced training set and testing set on which we compare GRIL with several commonly used GNN models from the literature including: Graph Convolutional Networks (GCN) (Kipf & Welling, 2017), Graph Isomorphism Networks (GIN) (Xu et al., 2019) and a 1-parameter persistent homology vector representation called persistence image (PersImg (Adams et al., 2017). All GNN models contain 3 aggregation layers. All models use 3-layer multilayer perceptron (MLP) as classifiers. More details about model and training settings can be found in Appendix C.1. We also test these trained models on HourGlass[10,20] and HourGlass[31,40] to check if they can generalize well on smaller and larger graphs. The experimental results are shown in Table 1. We can see that this dataset can be easily classified by our model based on 2-parameter persistence modules with good generalization performance but 1-parameter persistence method like PersImg or some GNN models struggle with this dataset.

5.2. Graph Experiments

We perform a series of experiments on graph classification to test the proposed model. We use standard datasets such as PROTEINS, DHFR, COX2, IMDB-BINARY and MUTAG (Morris et al., 2020). A quantitative summary of these datasets is given in Appendix C.2.

5.2.1. CLASSIFYING GRIL REPRESENTATIONS DIRECTLY

We compare the performance of GRIL with other models such as multiparameter persistence landscapes (MP-L) (Vipond, 2020), multiparameter persistence images (MP-I) (Carrière & Blumberg, 2020), multiparameter persistence kernel (MP-K) (Corbet et al., 2019).

In (Carrière & Blumberg, 2020), the authors use the heat kernel signature (HKS) and Ricci curvature to form a bifiltration on the graph datasets. We also use the same bifiltration and report the result in Table 2. We use XGBoost classifier (Chen & Guestrin, 2016) as done in (Carrière & Blumberg, 2020) for a fair comparison. We also report the results of GRIL with different classifiers in Table 9. The reported accuracies are averaged over 5 train/test splits of the datasets obtained with 5 stratified folds. The full details of the experiments are given in Appendix C.2.

From Table 2, we can see that the performance of GRIL on IMDB-BINARY is slightly lower than the other methods. This is because the graphs in IMDB-BINARY do not contain many cycles and hence, there is not enough information to capture in H_1 (See Appendix F for a visual interpretation). However, when there is information available, GRIL captures it better than the existing methods as can be seen from the accuracy values on other datasets.

5.2.2. AUGMENTING GNNS WITH GRIL FEATURES

Experimental Setup. In another set of experiments, we augment standard GNNs with GRIL features and compare the performance of the model with the existing ones. We use 3 layers of message-passing with hidden dimensionality of 64. The latent node representations are passed through a pooling layer and a two layer MLP to obtain the final classification. We use sum pooling to maintain uniformity among experiments and we do not claim that this is the optimal choice in any sense. For the GNN+GRIL architectures, we concatenate H_0 and H_1 and pass it through a 1-layer

Testing accuracy of models on HourGlass				
Model	**GCN**	**GIN**	**PersImg**	**GRIL**
HourGlass[21,30]	87.25±4.0	84.00±4.4	74.00±7.4	100.0±0.0
HourGlass[10,20]	67.31±4.6	62.98±3.4	50.33±1.6	99.79±0.1
HourGlass[31,40]	87.75±2.2	79.10±6.2	86.95±5.0	100.0±0.0

Table 1: Table of testing results from different models. Last two rows show the testing results on HourGlass[10,20] and HourGlass[31,40] of models trained on HourGlass[21,30]. For each dataset accuracies reported in red and blue denote the best and second-best performance respectively.

Dataset	**MP-I**	**MP-K**	**MP-L**	**P**	**GRIL**
PROTEINS	67.3 ± 3.5	67.5 ± 3.1	65.8 ± 3.3	65.4 ± 2.7	70.9 ± 3.1
DHFR	80.2 ± 2.3	81.7 ± 1.9	79.5 ± 2.3	70.9 ± 3.1	77.6 ± 2.5
COX2	77.9 ± 2.7	79.9 ± 1.8	79.0 ± 3.3	76.0 ± 4.1	79.8 ± 2.9
MUTAG	85.6 ± 7.3	86.2 ± 2.6	85.7 ± 2.5	79.2 ± 7.7	87.8 ± 4.2
IMDB-BINARY	71.1 ± 2.1	68.2 ± 1.2	71.2 ± 2.0	54.0 ± 1.9	65.2 ± 2.6

Table 2: Test accuracy of different models on graph datasets. The values of the MP-I, MP-K, MP-L and P columns are as reported in (Carrière & Blumberg, 2020); P denotes 1-parameter persistence.

Model	PROTEINS	DHFR	COX2	MUTAG	IMDB-BINARY
GCN	71.15 ± 2.31	78.70 ± 2.35	78.80 ± 2.13	88.26 ± 3.70	73.1 ± 2.20
GCN + GRIL	74.21 ± 2.08	75.66 ± 3.08	80.30 ± 1.57	88.80 ± 3.60	72.6 ± 1.46
GAT	67.66 ± 3.92	77.78 ± 4.50	79.45 ± 3.68	86.69 ± 6.36	74.90 ± 2.98
GAT + GRIL	71.60 ± 3.92	79.64 ± 6.29	80.52 ± 3.30	84.03 ± 7.85	71.60 ± 3.04
GIN	69.09 ± 3.77	79.77 ± 6.72	78.80 ± 4.88	83.97 ± 6.04	73.7 ± 3.34
GIN + GRIL	71.87 ± 3.22	78.46 ± 5.80	79.22 ± 4.89	89.32 ± 4.81	74.2 ± 2.82

Table 3: Performance comparison of baseline GNNs and GRIL augmented GNNs on graph benchmark datasets.

Model	IMDB-BINARY	IMDB-MULTI	REDDIT-BINARY	REDDIT-MULTI-5K
	initial_node_features: deg(v)		initial_node_features: uninformative	
GIN	73.70 ± 3.34	49.60 ± 3.02	90.30 ± 1.30	53.77 ± 1.85
GIN + GRIL	74.20 ± 2.82	50.33 ± 2.58	87.35 ± 2.77	53.85 ± 2.60

Table 4: Performance comparison of baseline GNNs and GRIL augmented GNNs on social network datasets without node attributes.

MLP. We concatenate the transformed GRIL values with the graph-level representations obtained from the pooling layer before passing through the final MLP classifier.

Training and evaluation. The models are trained for 100 epochs with ADAM as the optimizer. The initial learning rate was set to be 10^{-2} halving every 20 epochs. No hyperparameter tuning and early stopping was done. Though restrictive for practical scenarios, we follow earlier works (see (Morris et al., 2019),(Zhang et al., 2018) for more details). We report cross-validation accuracy averaged over 10 folds of the model obtained in the final training epoch.

Results. We can see from Table 3 that GRIL captures topological information that the GNN architectures are unable to capture and hence we see a clear increase in performance. However this is not the case for social network datasets. For the experiments reported in table 4 the node features are set as *uninformative* following the settings of (Xu et al., 2019). For the IMDB-*, REDDIT-MULTI-5K datasets, the augmented GRIL features improve the baseline GIN accuracy. For the REDDIT-BINARY dataset, since the graphs are highly sparse GRIL features computed with HKS-RC bifiltration fails to capture important features and as a consequence, the performance decreases.

6. Conclusions

In this work, we propose GRIL, a 2-parameter persistence vectorization based on generalized rank invariant that we show is Lipschitz continuous and differentiable with respect to the bi-filtration functions. Further, we present an algorithm for computing GRIL which is a synergistic confluence of the recent developments in computing generalized rank invariant of a 2-parameter module and an efficient algorithm for computing zigzag persistence. As a topological feature extractor, GRIL performs better than Graph Convolutional Networks (GCNs) and Graph Isomorphism Networks (GINs) on our synthetic dataset. It also performs better than the existing multiparameter persistence methods on some graph benchmark datasets while achieves comparable performance on others. Furthermore, our results indicate that GRIL may aid GNNs achieving better accuracies for graph classification. We believe that the additional topological information that a 2-parameter persistence module encodes, as compared to a 1-parameter persistence module, can be leveraged to learn better representations. Further directions of research include using GRIL with GNNs for filtration learning to learn more powerful representations. We expect that this work motivates further research in this direction

Acknowledgment

This research is partially supported by NSF grant CCF 2049010.

References

Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peterson, Patrick Shipman, Sofya Chepushtanova, Eric Hanson, Francis Motta, and Lori Ziegelmeier. Persistence images: A stable vector representation of persistent homology. *Journal of Machine Learning Research*, 18(8):1–35, 2017. URL http://jmlr.org/papers/v18/16-337.html.

Aaron Adcock, Daniel Rubin, and Gunnar Carlsson. Classification of hepatic lesions using the matching metric. *Comput. Vis. Image Underst.*, 121:36–42, apr 2014. ISSN 1077-3142. doi: 10.1016/j.cviu.2013.10.014. URL https://doi.org/10.1016/j.cviu.2013.10.014.

Josh Alman and Virginia Vassilevska Williams. A refined laser method and faster matrix multiplication. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 522–539, 2021. doi: 10.1137/1.9781611976465.32. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611976465.32.

Giorgos Bouritsas, Fabrizio Frasca, Stefanos P Zafeiriou, and Michael Bronstein. Improving graph neural network expressivity via subgraph isomorphism counting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2022. doi: 10.1109/TPAMI.2022.3154319. URL https://doi.org/10.1109/TPAMI.2022.3154319.

Peter Bubenik. Statistical topological data analysis using persistence landscapes. *J. Mach. Learn. Res.*, 16:77–102, 2015. doi: 10.5555/2789272.2789275. URL https://dl.acm.org/doi/10.5555/2789272.2789275.

Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pp. 108–122, 2013.

Zixuan Cang and Guo-Wei Wei. Topologynet: Topology based deep convolutional and multi-task neural networks for biomolecular property predictions. *PLOS Computational Biology*, 13(7):1–27, 07 2017. doi: 10.1371/journal.pcbi.1005690. URL https://doi.org/10.1371/journal.pcbi.1005690.

Gunnar Carlsson and Vin De Silva. Zigzag persistence. *Foundations of computational*

mathematics, 10(4):367–405, 2010. URL https://link.springer.com/content/pdf/10.1007/s10208-010-9066-0.pdf.

Gunnar Carlsson and Mikael Vejdemo-Johansson. *Topological Data Analysis with Applications*. Cambridge University Press, 2021. doi: 10.1017/9781108975704.

Gunnar Carlsson and Afra Zomorodian. The theory of multidimensional persistence. *Discrete & Computational Geometry*, 42(1):71–93, Jul 2009. ISSN 1432-0444. doi: 10.1007/s00454-009-9176-0. URL https://doi.org/10.1007/s00454-009-9176-0.

Mathieu Carrière and Andrew Blumberg. Multiparameter persistence image for topological machine learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 22432–22444. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/fdff71fcab656abfbefaabecab1a7f6d-Paper.pdf.

Mathieu Carrière, Frédéric Chazal, Yuichi Ike, Theo Lacombe, Martin Royer, and Yuhei Umeda. Perslay: A neural network layer for persistence diagrams and new graph topological signatures. In Silvia Chiappa and Roberto Calandra (eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 2786–2796. PMLR, 26–28 Aug 2020. URL https://proceedings.mlr.press/v108/carriere20a.html.

Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Frédéric Chazal, David Cohen-Steiner, Marc Glisse, Leonidas Guibas, and Steve Y. Oudot. Proximity of persistence modules and their diagrams. In *Proceedings of the Twenty-fifth Annual Symposium on Computational Geometry*, SCG '09, pp. 237–246, 2009a.

Frédéric Chazal, David Cohen-Steiner, Marc Glisse, Leonidas J. Guibas, and Steve Y. Oudot. Proximity of persistence modules and their diagrams. In *Proceedings of the Twenty-Fifth Annual Symposium on Computational Geometry*, SCG '09, pp. 237–246, New York, NY, USA, 2009b. Association for Computing Machinery. ISBN 9781605585017. doi: 10.1145/1542362.1542407. URL https://doi.org/10.1145/1542362.1542407.

Chao Chen, Xiuyan Ni, Qinxun Bai, and Yusu Wang. A topological regularizer for classifiers via persistent homology. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2573–2582. PMLR, 2019.

Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pp. 785–794, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939785. URL https://doi.org/10.1145/2939672.2939785.

René Corbet, Ulderico Fugacci, Michael Kerber, Claudia Landi, and Bei Wang. A kernel for multiparameter persistent homology. *Computers & Graphics: X*, 2:100005, 2019. ISSN 2590-1486. doi: https://doi.org/10.1016/j.cagx.2019.100005. URL https://www.sciencedirect.com/science/article/pii/S2590148619300056.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, Sep 1995. ISSN 1573-0565. doi: 10.1007/BF00994018. URL https://doi.org/10.1007/BF00994018.

Nima Dehmamy, Albert-László Barabási, and Rose Yu. *Understanding the Representation Power of Graph Neural Networks in Learning Graph Topology*. Curran Associates Inc., Red Hook, NY, USA, 2019.

Andac Demir, Baris Coskunuzer, Yulia Gel, Ignacio Segovia-Dominguez, Yuzhou Chen, and Bulent Kiziltan. ToDD: Topological compound fingerprinting in computer-aided drug discovery. In *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=8hs7qlWcnGs.

Tamal K. Dey and Tao Hou. Fast computation of zigzag persistence. In *30th Annual European Symposium on Algorithms, ESA 2022, September 5-9, 2022, Berlin/Potsdam, Germany*, volume 244 of *LIPIcs*, pp. 43:1–43:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi: 10.4230/LIPIcs.ESA.2022.43. URL https://doi.org/10.4230/LIPIcs.ESA.2022.43.

Tamal K. Dey and Yusu Wang. *Computational Topology for Data Analysis*. Cambridge University Press, 2022. doi: 10.1017/9781009099950.

Tamal K. Dey, Woojin Kim, and Facundo Mémoli. Computing generalized rank invariant for 2-parameter persistence modules via zigzag persistence and its applications. In Xavier Goaoc and Michael Kerber (eds.), *38th International Symposium on Computational Geometry, SoCG 2022, June 7-10, 2022, Berlin, Germany*, volume 224 of *LIPIcs*, pp. 34:1–34:17. Schloss

Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi: 10.4230/LIPIcs.SoCG.2022.34. URL https://doi.org/10.4230/LIPIcs.SoCG.2022.34.

Herbert Edelsbrunner and John Harer. *Computational Topology: An Introduction*. Applied Mathematics. American Mathematical Society, 2010. ISBN 9780821849255.

Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pp. 454–463, 2000. doi: 10.1109/SFCS.2000.892133.

Lawrence C. Evans and Ronald F. Gariepy. *Measure theory and fine properties of functions*. Textbooks in Mathematics. CRC Press, Boca Raton, FL, revised edition, 2015. ISBN 978-1-4822-4238-6.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *J. Mach. Learn. Res.*, 9:1871–1874, jun 2008. ISSN 1532-4435.

Robin Forman. Bochner's method for cell complexes and combinatorial Ricci curvature. *Discret. Comput. Geom.*, 29(3):323–374, 2003. doi: 10.1007/s00454-002-0743-x. URL https://doi.org/10.1007/s00454-002-0743-x.

Rickard Brüel Gabrielsson, Bradley J. Nelson, Anjan Dwaraknath, and Primoz Skraba. A topology layer for machine learning. In *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pp. 1553–1563. PMLR, 2020. URL http://proceedings.mlr.press/v108/gabrielsson20a.html.

M. Gori, G. Monfardini, and F. Scarselli. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pp. 729–734 vol. 2, 2005. doi: 10.1109/IJCNN.2005.1555942. URL https://doi.org/10.1109/IJCNN.2005.1555942.

Allen Hatcher. *Algebraic topology*. Cambridge Univ. Press, Cambridge, 2000. URL https://cds.cern.ch/record/478079.

Christoph D. Hofer, Roland Kwitt, Marc Niethammer, and Andreas Uhl. Deep learning with topological signatures. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 1634–1644, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/883e881bb4d22a7add958f2d6b052c9f-Abstract.html.

Christoph D. Hofer, Florian Graf, Bastian Rieck, Marc Niethammer, and Roland Kwitt. Graph filtration learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4314–4323. PMLR, 2020. URL http://proceedings.mlr.press/v119/hofer20b.html.

Max Horn, Edward De Brouwer, Michael Moor, Yves Moreau, Bastian Rieck, and Karsten M. Borgwardt. Topological graph neural networks. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=oxxUMeFwEHd.

Bryn Keller, Michael Lesnick, and Theodore L. Willke. Phos: Persistent homology for virtual screening. *ChemRxiv*, 2018. doi: 10.26434/chemrxiv.6969260.v1.

Kwangho Kim, Jisu Kim, Manzil Zaheer, Joon Kim, Frédéric Chazal, and Larry Wasserman. PLLay: Efficient topological layer based on persistent landscapes. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 15965–15977. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/b803a9254688e259cde2ec0361c8abe4-Paper.pdf.

Woojin Kim and Facundo Mémoli. Generalized persistence diagrams for persistence modules over posets. *Journal of Applied and Computational Topology*, 5(4):533–581, Dec 2021. ISSN 2367-1734. doi: 10.1007/s41468-021-00075-1. URL https://doi.org/10.1007/s41468-021-00075-1.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.

Michael Lesnick. The theory of the interleaving distance on multidimensional persistence modules. *Found. Comput. Math.*, 15(3):613–650, jun 2015. ISSN 1615-3375. doi: 10.1007/s10208-015-9255-y. URL https://doi.org/10.1007/s10208-015-9255-y.

Michael Lesnick and Matthew Wright. Interactive visualization of 2-d persistence modules. *CoRR*, abs/1512.00180, 2015. URL http://arxiv.org/abs/1512.00180.

Xiang Liu, Huitao Feng, Jie Wu, and Kelin Xia. Dowker complex based machine learning (dcml) models for protein-ligand binding affinity prediction. *PLOS Computational Biology*, 18(4):1–17, 04 2022. doi: 10.1371/journal.pcbi.1009943. URL https://doi.org/10.1371/journal.pcbi.1009943.

Saunders MacLane. *Categories for the working mathematician*. Graduate Texts in Mathematics, Vol. 5. Springer-Verlag, New York-Berlin, 1971.

Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 4602–4609, 2019.

Christopher Morris, Nils M. Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. TU-Dataset: A collection of benchmark datasets for learning with graphs. In *ICML 2020 Workshop on Graph Representation Learning and Beyond (GRL+ 2020)*, 2020. URL www.graphlearning.io.

Chien-Chun Ni, Yu-Yao Lin, Feng Luo, and Jie Gao. Community detection on networks with Ricci flow. *Scientific Reports*, 9(1):9984, Jul 2019. ISSN 2045-2322. doi: 10.1038/s41598-019-46380-9. URL https://doi.org/10.1038/s41598-019-46380-9.

Steve Y. Oudot. *Persistence Theory: From Quiver Representations to Data Analysis*. Number 209 in Mathematical Surveys and Monographs. American Mathematical Society, 2015. URL https://hal.inria.fr/hal-01247501.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf.

Amit Patel. Generalized persistence diagrams. *J. Appl. Comput. Topol.*, 1(3-4):397–419, 2018. doi: 10.1007/s41468-018-0012-6. URL https://doi.org/10.1007/s41468-018-0012-6.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009. doi: 10.1109/TNN.2008.2005605. URL https://doi.org/10.1109/TNN.2008.2005605.

Nicolas Swenson, Aditi S Krishnapriyan, Aydin Buluc, Dmitriy Morozov, and Katherine Yelick. PersGNN: Applying topological data analysis and geometric deep learning to structure-based protein function prediction. *arXiv preprint arXiv:2010.16027*, 2020.

Oliver Vipond. Multiparameter persistence landscapes. *Journal of Machine Learning Research*, 21(61):1–38, 2020. URL http://jmlr.org/papers/v21/19-054.html.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=ryGs6iA5Km.

Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. An end-to-end deep learning architecture for graph classification. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Simon Zhang, Soham Mukherjee, and Tamal K. Dey. GEFL: extended filtration learning for graph classification. In *Proceedings of the First Learning on Graphs Conference*, volume 198 of *Proceedings of Machine Learning Research*, pp. 16:1–16:26. PMLR, 09–12 Dec 2022. URL https://proceedings.mlr.press/v198/zhang22b.html.

Qi Zhao, Ze Ye, Chao Chen, and Yusu Wang. Persistence enhanced graph neural network. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 2896–2906. PMLR, 26–28 Aug 2020. URL https://proceedings.mlr.press/v108/zhao20d.html.

Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. *Discrete & Computational Geometry*, 33(2):249–274, Feb 2005. ISSN 1432-0444. doi: 10.1007/s00454-004-1146-y. URL https://doi.org/10.1007/s00454-004-1146-y.

A. Background and definitions

Here, we give the detailed definitions of all the concepts explained in the paper. We begin by defining a simplicial complex.

Definition A.1 (Simplicial Complex)**.** An abstract simplicial complex is a pair (V, Σ) where V is a finite set and Σ is a collection of non-empty subsets of V such that if $\sigma \in \Sigma$ and if $\tau \subseteq \sigma$ then $\tau \in \Sigma$. A topological space $|(V, \Sigma)|$ can be associated with the simplicial complex which can be defined using a bijection $t\colon V \to \{1, 2, \ldots, |V|\}$ as the subspace of $\mathbb{R}^{|V|}$ formed by the union $\bigcup_{\sigma\in\Sigma} h(\sigma)$, where $h(\sigma)$ denotes the convex hull of the set $\{e_{t(s)}\}_{s\in\sigma}$, where e_i denotes the standard basis vector in $\mathbb{R}^{|V|}$.

We shall now define a zigzag filtration and the zigzag persistence module induced by it.

Definition A.2. A *zigzag filtration* is a sequence of simplicial complexes where both insertions and deletions of simplices are allowed, the possibility of which we indicate with double arrows:

$$X_0 \leftrightarrow X_1 \leftrightarrow \cdots \leftrightarrow X_n = \mathcal{X}.$$

Applying homology functor on such a filtration we get a zigzag persistence module that is a sequence of vector spaces connected either by forward or backward linear maps:

$$H_*(X_0) \leftrightarrow H_*(X_1) \leftrightarrow \cdots \leftrightarrow H_*(X_n).$$

Now, we give the definition of 2-parameter filtration over $\mathbb{R}^2$ and the 2-parameter persistence module induced by it.

Definition A.3 (2-parameter simplicial filtration over $\mathbb{R}^2$)**.** A 2-parameter simplicial filtration, also called bi-filtration, over $\mathbb{R}^2$ is a collection of simplicial complexes $\{X_{\mathbf{u}}\}_{\mathbf{u}\in\mathbb{R}^2}$ with inclusion maps $X_{\mathbf{u}} \hookrightarrow X_{\mathbf{v}}$ for $\mathbf{u} \leq \mathbf{v}$, that is, $u_1 \leq u_2$ and $v_1 \leq v_2$ where $\mathbf{u} = (u_1, u_2)$ and $\mathbf{v} = (v_1, v_2)$.

Definition A.4 (2-parameter Persistence Module)**.** Given a bi-filtration, $\{X_{\mathbf{u}}\}_{\mathbf{u}\in\mathbb{R}^2}$, by considering the homology of the simplicial complexes in the bi-filtration over the finite field $\mathbb{Z}_2$, we get a collection of vector spaces $\{M_{\mathbf{u}} \mid \mathbf{u} \in \mathbb{R}^2\}$ along with a collection of linear maps $\{M_{\boldsymbol{u}\to\boldsymbol{v}} : M_{\mathbf{u}} \to M_{\mathbf{v}} \mid \boldsymbol{u} \leq \boldsymbol{v}\}$. Each inclusion map in the bi-filtration induces a linear map between the corresponding homology vector spaces.

Having defined 2-parameter filtration and 2-parameter persistence module, we now define the notion of an Interval in $\mathbb{R}^2$. In the definition, we shall make use of the standard partial order on $\mathbb{R}^2$, i.e., $\mathbf{u} \leq \mathbf{v}$ if $u_1 \leq v_1$ and $u_2 \leq v_2$ for $\mathbf{u} = (u_1, u_2)$ and $\mathbf{v} = (v_1, v_2)$.

Definition A.5. An interval in $\mathbb{R}^2$ is a subset $\emptyset \neq I \subseteq \mathbb{R}^2$ that satisfies the following:

1. If $\boldsymbol{u}, \boldsymbol{v} \in I$ and $\boldsymbol{u} \leq \boldsymbol{w} \leq \boldsymbol{v}$, then $\boldsymbol{w} \in I$;
2. If $\boldsymbol{u}, \boldsymbol{v} \in I$, then there exists a finite sequence $(\boldsymbol{u} = \boldsymbol{u}_0, \boldsymbol{u}_1, , ..., \boldsymbol{u}_m = \boldsymbol{v}) \in I$ so that every consecutive points $\boldsymbol{u}_i, \boldsymbol{u}_{i+1}$ are comparable in the partial order for $i \in \{0, \ldots, m-1\}$.

Let M be a 2-parameter persistence module over an interval $I \subseteq \mathbb{R}^2$. Then M admits a *limit* $\varprojlim M = (L, (\pi_{\boldsymbol{u}}\colon L \to M_{\boldsymbol{u}})_{\boldsymbol{u}\in I})$ and a *colimit* $\varinjlim M = (C, (i_{\boldsymbol{u}}\colon M_{\boldsymbol{u}} \to C)_{\boldsymbol{u}\in I})$ (Dey et al., 2022). Then, for every $\boldsymbol{u} \leq \boldsymbol{v}$, we have $M_{\boldsymbol{u}\to\boldsymbol{v}} \circ \pi_{\boldsymbol{u}} = \pi_{\boldsymbol{v}}$ and $i_{\boldsymbol{v}} \circ M_{\boldsymbol{u}\to\boldsymbol{v}} = i_{\boldsymbol{u}}$. This leads to $i_{\boldsymbol{u}} \circ \pi_{\boldsymbol{u}} = i_{\boldsymbol{v}} \circ \pi_{\boldsymbol{v}}$ which is a map from the limit L to the colimit C.

Definition A.6. (Kim & Mémoli, 2021) The *canonical limit-to-colimit* map for any such M is the map $\psi_M\colon \varprojlim M \to \varinjlim M$ given by $i_{\boldsymbol{v}} \circ \pi_{\boldsymbol{v}}$ for any $\boldsymbol{v} \in I$. The *generalized rank of M* is the rank of the map ψ_M (i.e. $\mathsf{rk}^M = \mathrm{rank}(\psi_M)$).

B. Stability and Differentiability: Proofs

In this section, we provide the proof for stability and differentiability of GRIL. We begin by introducing interleaving distance (Chazal et al., 2009b; Lesnick, 2015) and erosion distance (Patel, 2018; Kim & Mémoli, 2021) on the space of persistence modules.

Definition B.1. Given two persistence modules M and N, a morphism $f : M \to N$ is a collection of linear maps $\{f_{\boldsymbol{u}} : M_{\boldsymbol{u}} \to N_{\boldsymbol{u}}\}_{\boldsymbol{u}\in\mathbb{R}^2}$ such that $f_{\boldsymbol{u}} \circ N_{\boldsymbol{u}\to\boldsymbol{v}} = M_{\boldsymbol{u}\to\boldsymbol{v}} \circ f_{\boldsymbol{v}}, \forall \boldsymbol{u} \leq \boldsymbol{v}$.

Definition B.2. Given a persistence module M and $\epsilon \in \mathbb{R}$, we define the *shift module* $M^{\leftarrow\epsilon}$ through $M^{\leftarrow\epsilon}_{\boldsymbol{u}} = M_{\boldsymbol{u}+\epsilon}$ and $M^{\leftarrow\epsilon}_{\boldsymbol{u}\to\boldsymbol{v}} = M_{\boldsymbol{u}+\epsilon\to\boldsymbol{v}+\epsilon}$. Here $\boldsymbol{u} + \epsilon = (\boldsymbol{u}_1 + \epsilon, \boldsymbol{u}_2 + \epsilon)$.

Definition B.3. For a pair of persistence module M and N and some $\epsilon \in \mathbb{R}_{\geq 0}$, an ϵ-interleaving between M and N is a pair of morphisms $\phi : M \to N^{\leftarrow\epsilon}$ and $\psi : N \to M^{\leftarrow\epsilon}$ such that $\forall \boldsymbol{u} \in \mathbb{R}^2, M_{\boldsymbol{u}\to\boldsymbol{u}+2\epsilon} = \psi_{\boldsymbol{u}+\epsilon} \circ \phi_{\boldsymbol{u}}$ and $N_{\boldsymbol{u}\to\boldsymbol{u}+2\epsilon} = \phi_{\boldsymbol{u}+\epsilon} \circ \psi_{\boldsymbol{u}}$. If such interleaving exists, we say M and N are ϵ-interleaved.

Definition B.4. For two persistence modules M and N, we define the *interleaving distance* as $\mathrm{d_I}(M, N) \triangleq \inf_{\epsilon\geq 0}\{M \text{ and } N \text{ are } \epsilon\text{-interleaved}\}$.

Definition B.5. For persistence module M, N with GRILs λ^M, λ^N, define

$$d_{\mathcal{L}}(M, N) \triangleq ||\lambda^M - \lambda^N||_\infty.$$

Definition B.6. Given any interval I and $\varepsilon \geq 0$, let $I^{+\varepsilon}$ be the *ε-extension* of I defined as:

$$I^{+\varepsilon} \triangleq \bigcup_{\boldsymbol{p}\in I} \boxed{\boldsymbol{p}}_\varepsilon \tag{4}$$

where $\boxed{\boldsymbol{p}}_\varepsilon \triangleq \{\boldsymbol{q} : ||\boldsymbol{p}-\boldsymbol{q}||_\infty \leq \varepsilon\}$ is the ∞-norm ε-neighbourhood of x.

Based on generalized rank invariants over all intervals in $\mathbb{R}^2$, one can define erosion distance as follows:

Definition B.7. Let $\mathbf{Int}(\mathbb{R}^2)$ be the collection of all intervals in $\mathbb{R}^2$. Let M and N be two persistence modules. The *erosion distance* is defined as

$$\mathrm{d}_{\mathcal{E}}(M,N) \triangleq \inf_{\varepsilon\geq 0}\{\forall I \in \mathbf{Int}(\mathbb{R}^2), \\ \mathsf{rk}^M(I) \geq \mathsf{rk}^N(I^{+\varepsilon}) \text{ and } \mathsf{rk}^N(I) \geq \mathsf{rk}^M(I^{+\varepsilon})\}.$$

In order to better analyze the stability property of persistence landscape, we define a distance in a similar flavour as erosion distance for the underlying collection of all worms.

Definition B.8. For $\mathcal{W} \triangleq \left\{\boxed{\boldsymbol{p}}^\ell_\delta \mid \delta>0, l\in\mathbb{N}_+, \boldsymbol{p}\in\mathbb{R}^2\right\}$, define a distance $\mathrm{d}_{\mathcal{E}}^{\mathcal{W}}$ as follows:

$$\mathrm{d}_{\mathcal{E}}^{\mathcal{W}}(M,N) \triangleq \\ \inf_{\varepsilon\geq 0}\{\forall \boxed{\boldsymbol{p}}^\ell_\delta \in \mathcal{W}, [\mathsf{rk}^M\left(\boxed{\boldsymbol{p}}^\ell_\delta\right) \geq \mathsf{rk}^N\left(\boxed{\boldsymbol{p}}^\ell_{\varepsilon+\delta}\right) \text{ and } \\ \mathsf{rk}^N\left(\boxed{\boldsymbol{p}}^l_\delta\right) \geq \mathsf{rk}^M\left(\boxed{\boldsymbol{p}}^\ell_{\varepsilon+\delta}\right)]\}.$$

Proposition B.9. $\mathrm{d}_{\mathcal{L}} = \mathrm{d}_{\mathcal{E}}^{\mathcal{W}} \leq \mathrm{d}_{\mathcal{E}}$.

Proof. $\mathrm{d}_{\mathcal{E}}^{\mathcal{W}} \leq \mathrm{d}_{\mathcal{E}}$ is obvious by definition.

To show $\mathrm{d}_{\mathcal{L}} \leq \mathrm{d}_{\mathcal{E}}^{\mathcal{W}}$. Given two persistence modules M, N, assume $\mathrm{d}_{\mathcal{E}}^{\mathcal{I}}(M,N)=\epsilon$. For fixed $\boldsymbol{p}, k, \ell$, let $\lambda^M(\boldsymbol{p},k,\ell)=\delta_1$ and $\lambda^N(\boldsymbol{p},k,\ell)=\delta_2$. Without loss of generality, assume $\delta_2 \geq \delta_1$. We want to show that $\delta_2-\delta_1 \leq \epsilon$. By the construction of $\mathrm{d}_{\mathcal{E}}^{\mathcal{W}}$, we know that for any $\alpha>0$, $k > \mathsf{rk}^N(\boxed{\boldsymbol{p}}^\ell_{\delta_1+\alpha}(x)) \geq \mathsf{rk}^M(\boxed{\boldsymbol{p}}^\ell_{\delta_1+\epsilon+\alpha}(x))$. One can get $\delta_1+\epsilon+\alpha>\delta_2 \implies \epsilon+\alpha>\delta_2-\delta_1$. By taking $\alpha\to 0$, we have $\delta_2-\delta_1\leq\epsilon$.

To show $\mathrm{d}_{\mathcal{E}}^{\mathcal{W}} \leq \mathrm{d}_{\mathcal{L}}$. Let $\mathrm{d}_{\mathcal{L}}(M,N)=\delta$. For any $I=\boxed{\boldsymbol{p}}^\ell_\epsilon \in \mathcal{I}$, we want to show that $\mathsf{rk}^M(\boxed{\boldsymbol{p}}^\ell_\epsilon) \geq \mathsf{rk}^N(\boxed{\boldsymbol{p}}^\ell_{\epsilon+\delta})$ and $\mathsf{rk}^N(\boxed{\boldsymbol{p}}^\ell_\epsilon) \geq \mathsf{rk}^M(\boxed{\boldsymbol{p}}^\ell_{\epsilon+\delta})$. We prove the first inequality. The second one can be proved in a similar way. Let $k=\mathsf{rk}^N(\boxed{\boldsymbol{p}}^\ell_{\epsilon+\delta})$, then $\lambda^N(\boldsymbol{p},k,\ell)\geq\epsilon+\delta$. By the assumption $\mathrm{d}_{\mathcal{L}}(M,N)=\delta$, we know that $\lambda^N(\boldsymbol{p},k,\ell)\geq\epsilon$, which implies $\mathsf{rk}^M(\boxed{\boldsymbol{p}}^\ell_\epsilon)\geq k=\mathsf{rk}^N(\boxed{\boldsymbol{p}}^\ell_{\epsilon+\delta})$. □

Proposition 3.3. GRIL is equivalent to the generalized rank invariant on $\mathcal{W}$. Here equivalence means bijective reconstruction from each other.

Proof. Constructing GRIL from generalized rank invariant on $\mathcal{W}$ is immediate from the definition of GRIL.

On the other direction, for any $\boldsymbol{p}, \delta, \ell$, the generalized rank $\mathsf{rk}^M_{\mathcal{W}}(\boxed{\boldsymbol{p}}^\ell_\delta)$ can be reconstructed by GRIL as follows:

$$\mathsf{rk}^M_{\mathcal{W}}(\boxed{\boldsymbol{p}}^\ell_\delta) = \arg\max_k\{\lambda(\boldsymbol{p},k,\ell)\geq\delta\} \tag{5}$$

It is not hard to check that, this construction, combined with the construction of persistence landscape, gives a bijective mapping between (generalized) rank invariants over $\mathcal{W}$ and GRILs. □

By the stability property of erosion distances, we can immediately get the stability of GRIL as follows:

Proposition 3.4. For two filtration functions $f, f' : \mathcal{X}\to\mathbb{R}^2$, $||\lambda^{M^f}-\lambda^{M^{f'}}||_\infty \leq ||f-f'||_\infty$.

Proof. Let M^f and $M^{f'}$ be the persistence modules derived by f and f' respectively. Then, we have the following chain of inequalities:

$$\begin{aligned}\|\lambda^{M^f}-\lambda^{M^{f'}}\|_\infty =& \mathrm{d}_{\mathcal{L}}(M^f,M^{f'}) \\ \leq& \mathrm{d}_{\mathcal{E}}(M^f,M^{f'}) \\ \leq& d_I(M^f,M^{f'}) \\ \leq& \|f-f'\|_\infty\end{aligned}$$

where $\mathrm{d}_{\mathrm{I}}(M^f,M^{f'})$ is the interleaving distance. The second last inequality has been shown in (Kim & Mémoli, 2021). □

Recall that when $\mathcal{X}$ is a finite space (e.g. finite simplicial complex, point cloud) then, any $f:\mathcal{X}\to\mathbb{R}^2$ can be considered as an $n\times 2$ matrix which can be linearized into a vector in $\mathbb{R}^{2n}$. Let us denote that vector by v_f.

C. Experimental Setup

C.1. Hourglass Dataset

The two traversals T_1 and T_2 are designed as follows: T_1 traverses G_1, then followed by G_2; T_2 traverses upper halves $G_1^\top \subseteq G_1$ and $G_2^\top \subseteq G_2$ sequentially first, then followed by the other halves $G_1^\perp \subseteq G_1$ and $G_2^\perp \subseteq G_2$. For cross edges, we randomly pick $2|V|$ pairs of nodes (with replacement) in $G_1^\top \times G_2^\perp$ on which we place cross edges. We don't place multiple edges on the same pair of nodes. In a similar way we place cross edges on $G_1^\perp \times G_2^\top$. Therefore, G has roughly $6|V|$ cross edges between G_1 and G_2. The (roughly) total number of edges: $|E|\approx 5|V|$. For methods based on persistence modules, we take two filtration functions $f_1, f_2 : V\cup E\to\mathbb{R}$ on G as follows: let $x(v)$ be the node attribute on v given by the order index of the trace. Then

- f_1 is given by $\forall v\in V, f_1(v)=x(v)$ and $\forall e=(v,w)\in E, f_1(e)=\max(x(v),x(w))$.

- f_2 is given by $f_2(v) = 0$ and $f_2 = C(e)$ where $C(e)$ is a curvature value of e. Here we use a version of discrete Ricci called Forman-Ricci curvature (Forman, 2003) computed by the code provided in (Ni et al., 2019).

We compute for all points $\boldsymbol{p}$ in a uniform 4×4 grid the GRIL values $\lambda(\boldsymbol{p}, k, \ell)$ for generalized rank $k = 1, 2$, worm size $\ell = 2$, and homology of dimension 0 and 1. Therefore, for each graph our $\Lambda_{\boldsymbol{p}}^{k,\ell}$ generates a 64-dimensional vector as representation. For the method based on 1-parameter persistence modules with persistence image vectorization, we compute 1-parameter persistence modules for homology dimension $0, 1$ on f_1 and f_2 independently. Each persistence module will be vectorized on a 4×4 grid. Therefore, it also produces a 64-dimensional vector as representation.

C.2. Graph Experiments

We performed a series of experiments on graph classification using GRIL. We used standard datasets with node features such as PROTEINS, DHFR, COX2, MUTAG and IMDB-BINARY (Morris et al., 2020). Description of the graph classification tasks is given in Table 5.

Dataset	Num Graphs	Num Classes	Avg. No. Nodes	Avg. No. Edges
PROTEINS	1113	2	39.06	72.82
COX2	467	2	41.22	43.45
DHFR	756	2	42.43	44.54
MUTAG	188	2	17.93	19.79
IMDB-BINARY	1000	2	19.77	96.33

Table 5: Description of Graph Datasets

The Heat Kernel Signature-Ricci Curvature bi-filtration, as done in (Carrière & Blumberg, 2020), values are normalized so that they lie between 0 and 1. For the experiments reported in Section 5, we fix the grid resolution $\rho = 0.01$. Thus, the square $[0, 1] \times [0, 1]$ has 100×100 many grid points. We sample a uniform subgrid of center points, $\boldsymbol{p}$, out of these grid points. We fix $l = 2$ for our experiments. We compute $\lambda(\boldsymbol{p}, k, \ell)$ where $\boldsymbol{p}$ varies over the sampled center points and k varies from 1 to 5. Each such computation is done for dimension 0 homology (H_0) and dimension 1 homology (H_1). We use XGBoost (Chen & Guestrin, 2016) classifier for these experiments.

Ablation Studies. We have performed experiments with different subgrid sizes and the results are reported in Table 6. The reported accuracies are averaged over 5 train/test splits of the datasets obtained with 5 stratified folds. We can see from the table that for different datasets, different subgrid sizes give the best results. This can be attributed to the fact that for some datasets, topological information needs to be captured at a finer level while for other datasets, capturing such finer details can be redundant.

Grid Size	50×50	25×25	10×10	5×5
PROTEINS	70.8 ± 2.7	70.2 ± 1.8	69.8 ± 2.4	68.5 ± 2.6
DHFR	77.6 ± 2.5	77.2 ± 3.4	77.5 ± 3.5	77.5 ± 3.5
COX2	79.8 ± 3.0	78.9 ± 2.4	79.8 ± 2.9	78.9 ± 3.5
MUTAG	87.3 ± 3.8	87.8 ± 4.2	87.8 ± 4.5	86.8 ± 3.3
IMDB-BINARY	62.2 ± 4.3	65.2 ± 2.6	62.2 ± 2.3	63.5 ± 3.2

Table 6: Test accuracies of GRIL on subgrids of different sizes.

We report the computation times of GRIL for these datasets in Table 7. The values denote the total computation time for all the center points on a 50×50 subgrid for a 2-worm. The computations were done on a Intel(R) Xeon(R) Gold 6248R CPU machine and the computation was carried out on 32 cores.

Dataset	Computation time
PROTEINS	6 hr 13 min 38 s
DHFR	4 hr 15 min 54 s
COX2	2 hr 44 min 23 s
MUTAG	0 hr 56 min 48 s
IMDB-BINARY	4 hr 03 min 35 s

Table 7: Computation times for GRIL for each dataset with a 2-worm and 50×50 subgrid.

In Table 8, we show the performance of GRIL with different grid resolutions (ρ) and ℓ-worms. For these experiments, we used a 50×50 subgrid for the center points. The reported accuracies are averaged over 5 train/test splits of the datasets obtained with 5 stratified folds. We test it on MUTAG and COX2 and we can see that for $\rho = 0.01$, we get the highest accuracy of the model on both the datasets. We can see from the table that there is an improvement in accuracy from $\ell = 1$ to $\ell = 2$. However, there is no significant improvement from $\ell = 2$ to $\ell = 3$.

In Table 9, we report the performance of GRIL on graph benchmark datasets with different classifiers such as Support Vector Machine (SVM) (Cortes & Vapnik, 1995; Chang & Lin, 2011), Logistic Regression (LR) (Fan et al., 2008), Multilayer Perceptron (3-MLP) implemented using *scikit-learn* (Buitinck et al., 2013) library. The reported accuracies are averaged over 5 train/test splits of the datasets obtained with 5 stratified folds.

D. Algorithm

Here, we describe the algorithm in detail. In practice, we are usually presented with a piecewise linear (PL) approximation $\hat{f}$ of a $\mathbb{R}^2$-valued function f on a discretized domain such as a finite simplicial complex. The PL-approximation $\hat{f}$ itself is $\mathbb{R}^2$-valued. Discretizing the parameter space $\mathbb{R}^2$

Dataset	$\rho = 0.02$	$\rho = 0.01$	$\rho = 0.005$	$\ell = 1$	$\ell = 2$	$\ell = 3$
MUTAG	86.3 ± 4.2	87.8 ± 4.5	85.2 ± 3.9	85.7 ± 4.2	87.8 ± 4.5	87.8 ± 3.9
COX2	78.2 ± 1.7	79.8 ± 2.9	77.8 ± 1.4	79.3 ± 2.9	79.8 ± 2.9	78.9 ± 3.5

Table 8: Test accuracy for different grid resolutions and for ℓ-worms with different values of ℓ.

Dataset	SVM	LR	XGBoost	3-MLP
PROTEINS	73.3 ± 1.5	72.7 ± 2.6	70.9 ± 3.1	71.3 ± 2.1
DHFR	61.7 ± 0.4	77.8 ± 1.9	77.6 ± 2.5	72.3 ± 4.3
COX2	77.2 ± 0.8	78.5 ± 2.5	79.8 ± 2.9	77.0 ± 1.2
MUTAG	80.0 ± 3.9	86.3 ± 3.8	87.8 ± 4.2	76.8 ± 9.1
IMDB-BINARY	65.1 ± 3.6	63.2 ± 2.1	65.2 ± 2.6	61.2 ± 6.6

Table 9: Test accuracies of GRIL using different classifiers.

by a grid, we consider a *lower star* bi-filtration of the simplicial complex. Analogous to the 1-parameter case, a lower star bi-filtration is obtained by assigning every simplex the maximum of the values over all of its vertices in each of the two co-ordinates. With appropriate scaling, these (finite) values can be mapped to a subset of points in a uniform finite grid over $[0, 1] \times [0, 1]$. Observe that because of the maximization of values over all vertices, we have the property that two simplices $\sigma \subseteq \tau$ have values $\hat{f}(\sigma) \in \mathbb{R}^2$ and $\hat{f}(\tau) \in \mathbb{R}^2$ where $\hat{f}(\sigma) \leq \hat{f}(\tau)$. A partial order of the simplices according to these values provide a bi-filtration over the grid $[0, 1] \times [0, 1]$.

Computing generalized ranks. We need to compute the generalized rank $\mathsf{rk}^M(\widehat{\boxed{\boldsymbol{p}}}^{\ell}_d)$ for every worm $\widehat{\boxed{\boldsymbol{p}}}^{\ell}_d$ to decide whether to increase its width or not. We use a result of (Dey et al., 2022) to compute $\mathsf{rk}^M(\widehat{\boxed{\boldsymbol{p}}}^{\ell}_d)$. It says that $\mathsf{rk}^M(\widehat{\boxed{\boldsymbol{p}}}^{\ell}_d)$ can be computed by considering a zigzag module and computing the number of full bars (bars that begin at the start of the zigzag filtration and persist until the end of the filtration) in its decomposition. This zigzag module decomposition can be obtained by restricting the bi-filtration on the boundary of $\mathsf{rk}^M(\widehat{\boxed{\boldsymbol{p}}}^{\ell}_d)$ and using any of the zigzag persistence algorithms on the resulting zigzag filtration. We use the recently published efficient algorithm and its associated software (Dey & Hou, 2022) for computing zigzag persistence.

Computing the value of GRIL using binary search. For a worm $\widehat{\boxed{\boldsymbol{p}}}^{\ell}_d$ and a given $k \geq 1$, we apply binary search to compute the value of GRIL. Let us denote the grid resolution by ρ. We do the binary search for d in the range $[d_{\min}, d_{\max}]$ where $d_{\min} = \rho$ and $d_{\max} = 1$. In each iteration, we compute $\mathsf{rk}^M(\widehat{\boxed{\boldsymbol{p}}}^{\ell}_d)$ for $d = (d_{\min} + d_{\max})/2$ and check if $\mathsf{rk}^M(\widehat{\boxed{\boldsymbol{p}}}^{\ell}_d) \geq k$. We increase the width of the worm by updating $d_{\min}$ to be $d + \rho$ if $\mathsf{rk}^M(\widehat{\boxed{\boldsymbol{p}}}^{\ell}_d) \geq k$. Otherwise, we decrease the width of the worm by updating $d_{\max}$ to be $d - \rho$. The binary search stops and returns d when $d_{\max} < d_{\min}$. This ensures that we have searched through all possible values of d for which $\mathsf{rk}^M(\widehat{\boxed{\boldsymbol{p}}}^{\ell}_d) \geq k$ and returned the maximum of these values.

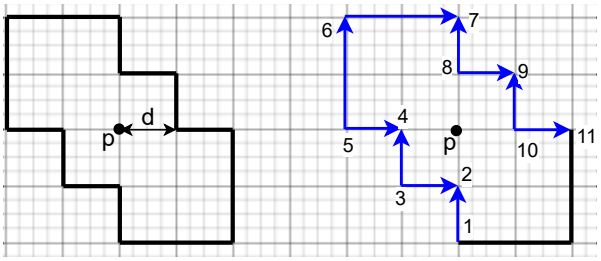

Figure 7: (Left) The figure shows the 2-worm centered at p with width d. (Right) The highlighted part denotes the boundary cap of the worm. The arrows in the figure denote the direction of arrows in the zigzag filtration.

Refer to Figure 7 for an illustration of the zigzag filtration along the boundary cap of a 2-worm.

E. More Discussion on Differentiability

Recall that the function function $\Lambda^{k,\ell}_{\boldsymbol{p}} : \mathbb{R}^{2n} \to \mathbb{R}$ is given by $\Lambda^{k,\ell}_{\boldsymbol{p}}(f) = \lambda^{M^f}(k, \ell, \boldsymbol{p})$. The differentiability of $\Lambda^{k,\ell}_{\boldsymbol{p}}$ in Corollary 3.6 refers to the existence of all directional derivatives. But the existence of a steepest direction as the "gradient" of $\Lambda^{k,\ell}_{\boldsymbol{p}}$ might not be unique. Here we propose an algorithm to efficiently compute one specific steepest direction based on the following theorem.

Theorem E.1. *Consider the space of all filtration functions $\{f : \mathcal{X} \to \mathbb{R}^2\}$ on a finite space $\mathcal{X}$ with $|\mathcal{X}| = n$, which is equivalent to $\mathbb{R}^{2n}$. For fixed $k, \ell, \boldsymbol{p}$, there exists a measure-zero subset $Z \subseteq \mathbb{R}^{2n}$ such that for any $f \in \mathbb{R}^{2n} \setminus Z$ satisfying the following generic condition: $\forall x \neq y \in \mathcal{X}, f(x)_1 \neq f(y)_1, f(x)_2 \neq f(y)_2$, there exists*

an assignment $s : \mathcal{X} \to \{\pm 1, 0, \pm\ell\}^2$ *such that*

$$\nabla_s \Lambda_{\boldsymbol{p}}^{k,\ell}(f) \triangleq \lim_{\alpha \to 0} \frac{\Lambda_{\boldsymbol{p}}^{k,\ell}(f + \alpha s) - \Lambda_{\boldsymbol{p}}^{k,\ell}(f)}{\alpha \|s\|_\infty} = \max_{g \in \mathcal{X}} \nabla_g \Lambda_{\boldsymbol{p}}^{k,\ell}(f).$$

Proof. By Corollary 3.6 we know there exists some measure-zero set $R \subset \mathbb{R}^{2n}$ such that $\Lambda_{\boldsymbol{p}}^{k,\ell}$ is differentiable in $\bar{R} \triangleq \mathbb{R}^{2n} \setminus R$. Let $M = M^f$ be a 2-parameter persistence module induced from some generic filtration function $f \in \bar{R}$ and $I = \boxed{\boldsymbol{p}}_d^\ell$ be an ℓ-worm in $\mathbb{R}^2$ centered at some point $\boldsymbol{p}$. Let $\partial(I)$ be the boundary of I excluding the right most vertical edge and bottom most horizontal edge (See Figure 8 as an illustration). It is shown in (Dey et al., 2022) that, over the boundary $\partial(I)$, a zigzag persistence module can be defined by restricting M to $\partial(I)$ (in practice it is enough to take a zigzag path to approximate the smooth off-diagonal boundary) on which the number of full bars is equal to $\mathsf{rk}^M(I)$. Let $I' = \boxed{\boldsymbol{p}}_{d'}^\ell$ be another ℓ-worm centered at $\boldsymbol{p}$ for some $d' \neq d$. One can observe that, if the zigzag filtrations on $\partial(I)$ and $\partial(I')$ have the same order of insertion and deletion of simplices , then the number of full bars on $M|_{\partial(I)}$ and $M|_{\partial(I')}$ are the same, which means $\mathsf{rk}^M(I) = \mathsf{rk}^M(I')$. Now let $d = \lambda^M(k, \ell, \boldsymbol{p}), I = \boxed{\boldsymbol{p}}_d^\ell, I_- = \boxed{\boldsymbol{p}}_{d-\varepsilon}^\ell, I_+ = \boxed{\boldsymbol{p}}_{d+\varepsilon}^\ell$ for some small enough ε. Based on the definition of λ^M, we know that $\mathsf{rk}^M(I_-) \geq k$ and $\mathsf{rk}^M(I_+) < k$, which means that zigzag filtrations change on some simplices while moving from $\partial(I_-)$ to $\partial(I_+)$. Either the collection of simplices changes or the order of simplices changes. The former case corresponds to the simplices with x or y-coordinate aligned with some vertical or horizontal edges on $\partial(I)$. The latter case corresponds to those pairs of simplices (σ, τ) such that $f(\sigma) \vee f(\tau) \triangleq (\max(f(\sigma)_1, f(\tau)_1), \max(f(\sigma)_2, f(\tau)_2)$ is on some off-diagonal edges on $\partial(I)$. By the generic condition of the filtration function f, we can locate those simplices as the set S, which we call support simplices. The assignment function s is defined on each $\sigma \in S$ by assigning $s(\sigma) = \pm 1$ or $\pm\ell$ which is consistent with the moving direction of the edge from $\partial(I)$ to $\partial(I_+)$. We discuss the assignment values case by case:

We can divide the boundary into four edges: bottom (off-diagonal) edge e_b, top (horizontal) edge e_t, left (vertical) edge e_l, right (off-diagonal) edge e_r.

1. $s(\sigma) = (0, +\ell)$ if σ has y-coordinate the same as e_t,
2. $s(\sigma) = (-\ell, 0)$ if σ has x-coordinate the same as e_l,
3. $s(\sigma) = (0, -1), s(\tau) = (-1, 0)$ if $f(\sigma) \vee f(\tau)$ is on e_b and $f(\sigma)_1 \leq f(\tau)_1$,
4. $s(\sigma) = (0, +1), s(\tau) = (+1, 0)$ if $f(\sigma) \vee f(\tau)$ is on e_r and $f(\sigma)_1 \leq f(\tau)_1$,

See Figure 8 as an illustration. We assume f satisfies the condition that the supporting simplices in S either all belong to cases 1 and 2 or all belong to cases 3 and 4, but not a combination of them. It is not hard to see that the collection of f for which this condition does not hold is a measure zero set in $\mathbb{R}^{2n}$. Let us denote the collection of all such f's by F. Then, $Z = F \cup R$ is a measure zero set in $\mathbb{R}^{2n}$ which consists of f's which do not satisfy the condition and those points where $\Lambda_{\boldsymbol{p}}^{k,\ell}$ is not differentiable.

Now, check for such a generic $f \notin Z$ so that the directional derivative $\nabla_s \lambda(f)$ is indeed a maximal directional derivative. For the cases 3 and 4, the stability property in Proposition 3.4 implies that, for any $\alpha > 0$ and any direction vector $g \in \mathbb{R}^{2n}$ with $\|g\|_\infty = 1$, we have $\lambda(f + \alpha g) - \lambda(f) \leq \alpha$. Also it is not hard to check that $\lambda(f + \alpha s) - \lambda(f) = \alpha$ for $\alpha > 0$ small enough since the zigzag persistence of $M^{f+\alpha s}|_J$ with $J = \boxed{\boldsymbol{p}}_{d+\alpha}^\ell$ has the same collection of simplices and orders as $M^f|_I$ with $I = \boxed{\boldsymbol{p}}_d^\ell$, which means they have the same rank. Therefore, we have $\forall \|g\|_\infty = 1, \lambda(f + \alpha g) - \lambda(f) \leq \lambda(f + \alpha s) - \lambda(f) \implies \nabla_g \Lambda(f) \leq \nabla_s \Lambda(f)$. For the case 1 (the case 2 is similar), the support simplex is on edge e_t. Now for any direction vector $g \in \mathbb{R}^{2n}$ and $\alpha > 0$ small enough, let $\Delta d = \Lambda(f + \alpha g) - \Lambda(f)$ and let Δy_{e_t} be the difference between y-coordinates of e_t's from $\boxed{\boldsymbol{p}}_d^\ell$ and $\boxed{\boldsymbol{p}}_{d+\Delta d}^\ell$. Note that $\frac{\Delta d}{\Delta y_{e_t}} = \ell$ and $\frac{|\Lambda(f+\alpha g) - \Lambda(f)|}{\alpha \|g\|_\infty} \leq \frac{\Delta d}{\Delta y_{e_t}}$ since in order to change $\Lambda(f)$ by Δd one has to at least move edge e_t by Δy_{e_t}, which correspondingly changes the y-coordinate of $s(\sigma)$ by Δy_{e_t}. From the above argument, we can get the directional derivative $\nabla_g \Lambda(f)$ is bounded from above by the ratio $\frac{\Delta d}{\Delta y_{e_t}} = \frac{1}{\ell} = \nabla_s \Lambda(f)$. The case for $\alpha < 0$ is symmetric.

In summary, $\nabla_s \lambda(f)$ indeed maximizes the directional derivative for f. □

The proof of Theorem E.1 also shows how to find the assignment s with the corresponding set of *supporting* simplices. This result enables us to update the filtration function of the simplices according to some target function based on $\Lambda_{\boldsymbol{p}}^{k,\ell}$. Here we introduce an experiment, as a proof of concept, to show how one can use GRIL as a machine learning model to enhance topological features. By giving a suitable target function, our model is trained to rearrange the positions of input points to better represent circles. The experiment results is shown in Figure 9. The input to $\Lambda_{\boldsymbol{p}}^{k,\ell}$ is points sampled non-uniformly from two circles. Recall that GRIL is defined over a 2-parameter persistence module induced by some filtration function $f = (f_x, f_y)$. For every vertex v, we assign $f_x(v) = 1 - \exp(\frac{1}{\alpha} \sum_{i=1}^{\alpha} d(v, v_i))$, where v_i denotes i-th nearest neighbor of the vertex v and $d(v, v_i)$ denotes the distance between v and v_i. For our

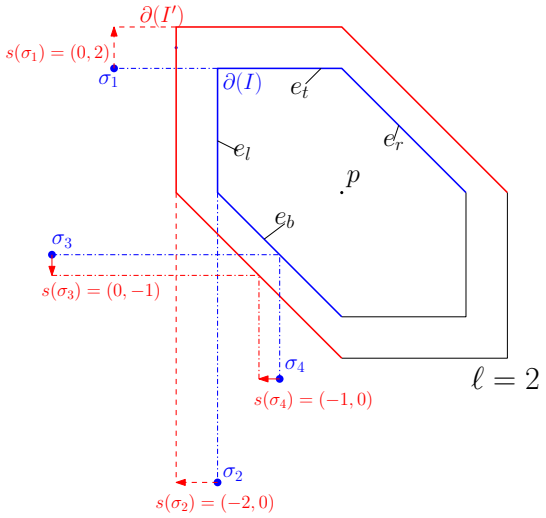

Figure 8: Two examples of 2-worm I, I'. Blue and red lines are boundaries of I and I' respectively on which the zigzag persistence modules are constructed for computing ranks. $\sigma_i, i = 1, 2, 3, 4$ are four support simplices on $\partial(I)$. $s(\sigma_i)$ is the assignment function values on σ_i.

experiments we fix $\alpha = 5$. We set $f_y(v) = 0$. We compute ALPHACOMPLEX filtration (Edelsbrunner & Harer, 2010) of the points and for each edge $e := (u, v)$ we assign $f_x(e) = \max(f_x(u), f_x(v))$ and $f_y(e) = 1 - \exp(d(u, v))$. To obtain a valid bi-filtration function on the simplicial complex we extend the bi-filtration function from 1-simplices to 2-simplices, i.e. triangles. We pass f as an input to $\Lambda_{\boldsymbol{p}}^{k,\ell}$, coded with the framework PYTORCH (Paszke et al., 2019), that computes persistence landscapes. $\Lambda_{\boldsymbol{p}}^{k,\ell}$ uniformly samples n center points from the grid $[0, 1]^2$. Since GRIL value computation can be done independently for each k and a center point, we take advantage of parallel computation and implement the code in a parallel manner. In the forward pass we get GRIL values $\lambda(\boldsymbol{p}, k, \ell)$ for generalized rank $k = 1, 2$, worm size $\ell = 2$ and homology of dimension 1 while varying $\boldsymbol{p}$ over all the sampled center points. After we get the GRIL values, we compute the assignment s according to Theorem E.1. During the backward pass, we utilize this assignment to compute the derivative of $\Lambda_{\boldsymbol{p}}^{k,\ell}$ with respect to the filtration function and consequently update it. We get n values of $\lambda(\cdot, 1, 2)$ for n center points. We treat these n values as a vector and denote is as λ_1. Similarly, we use λ_2 to denote the vector formed by values $\lambda(\cdot, 2, 2)$. We minimize the loss $L = -(\|\lambda_1\|_2^2 + \|\lambda_2\|_2^2)$. Figure 9 shows the result after running $\Lambda_{\boldsymbol{p}}^{k,\ell}$ for 200 epochs. The optimizer we use to optimize the loss function is Adam (Kingma & Ba, 2015) with a learning rate of 0.01.

F. Visualization of GRIL for graph datasets

The plot for first 5 GRIL values are shown in Figure 10. The figure contains landscape values for 5 random graph samples of each dataset. In Figure 11, we plot the first two eigen vectors given by principal component analysis (PCA) of the computed GRIL values for each dataset. Plots for H_0 and H_1 are shown separately.

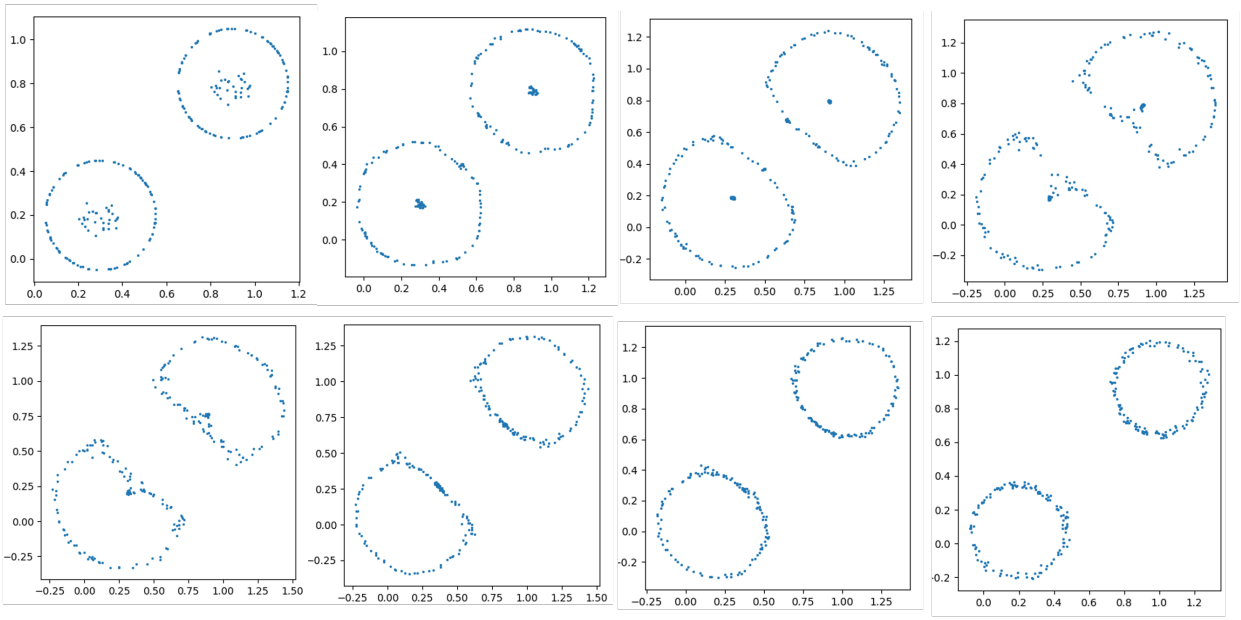

Figure 9: The figures show the rearrangement of points according to the loss function, which in our case is increasing the norm of λ_1 and λ_2 vectors. We start with two circles containing some noisy points inside. We observe that the points rearrange to form two circles because that increases the norm of λ_1 and λ_2 vectors.

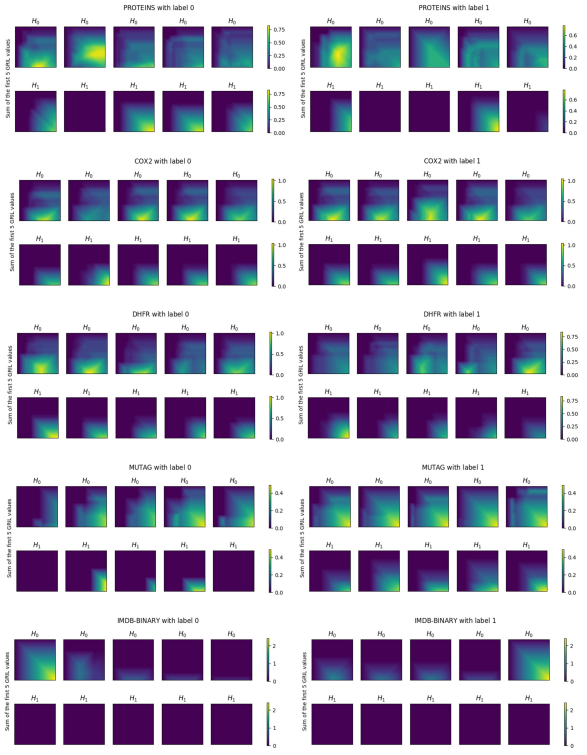

Figure 10: GRIL of 5 random graph samples of each dataset. GRIL values of H_0 and H_1 are shown separately column-wise.

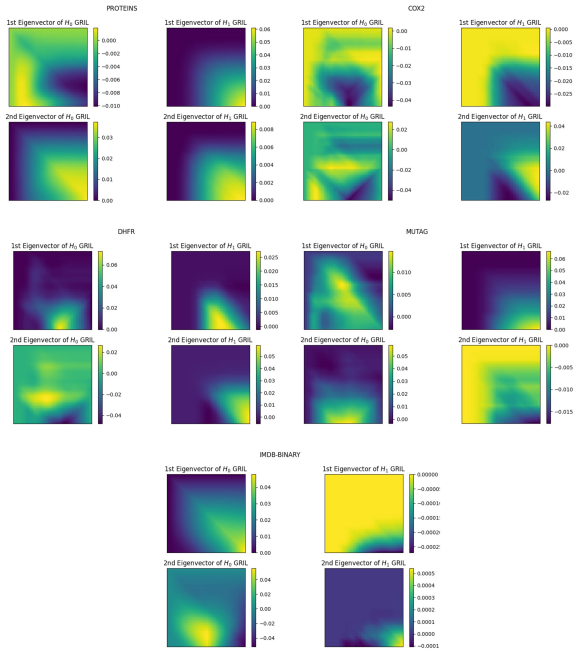

Figure 11: Plot of the first two eigen vectors given by PCA on the entire dataset for H_0 and H_1 respectively.