Quasi Zigzag Persistence: A Topological Framework for Analyzing Time-Varying Data

Tamal K. Dey*
Department of Computer Science
Purdue University
West Lafayette, IN
tamaldey@purdue.edu

Shreyas N. Samaga*
Department of Computer Science
Purdue University
West Lafayette, IN
ssamaga@purdue.edu

Abstract

In this paper, we propose Quasi Zigzag Persistent Homology (QZPH) as a framework for analyzing time-varying data by integrating multiparameter persistence and zigzag persistence. To this end, we introduce a stable topological invariant that captures both static and dynamic features at different scales. We present an algorithm to compute this invariant efficiently. We show that it enhances the machine learning models when applied to tasks such as sleep-stage detection, demonstrating its effectiveness in capturing the evolving patterns in time-varying datasets.

1 Introduction

Time varying data analysis Hamilton (1994); Box & Jenkins (1976) has been a fundamental challenge in the machine learning community, ranging from traditional time-series data to more complex structures such as sequences of graphs or point clouds. While traditional time-series have been handled effectively Wu et al. (2020), the complexity of modern applications necessitates novel methodologies. Spatiotemporal Graph Neural Networks Yu et al. (2018); Oreshkin et al. (2021); Kan et al. (2022); Chu et al. (2023) and specialized architectures for point cloud sequences Liu et al. (2019); Fan et al. (2021); Huang et al. (2021); Rempe et al. (2020) have emerged as powerful tools. However, most of these approaches utilize local geometric information, potentially missing crucial global patterns. This is particularly apparent where the overall shape carries significant meaning, such as in brain connectivity patterns. Topological methods offer a compelling solution by capturing these global, multi-scale structures. By augmenting models with topological information, we can leverage both local geometric patterns and global characteristics for improved performance.

Recently, Topological Data Analysis (TDA) has become a prominent field for leveraging such hidden information. Persistent Homology (PH), a cornerstone of TDA, provides a succinct method to extract *multiscale* topological features. This has been transformative in enhancing machine learning models. Analyzing time-varying data requires extending classical PH in two directions. First, the temporal component necessitates an additional parameter, bringing multiparameter persistence homology (MPH) Botnan & Lesnick (2023) into the picture. Second, standard PH computations use monotone filtrations which cannot accommodate the deletions required for time-varying data. This calls for zigzag persistent homology (ZPH) Carlsson & de Silva (2010), which captures dynamic topological features in time-series data Myers et al. (2023); Chen et al. (2021); Tinarrage et al. (2025); Hacquard & Lebovici (2024); Coskunuzer et al. (2024); Beltramo et al. (2022). Effectively, we need a combination of MPH and ZPH, allowing standard PH filtration along one parameter and ZPH along another. This requires structuring the underlying partially-ordered set (poset) as a *quasi-zigzag poset*, leading to what we term *Quasi Zigzag Persistent Homology (QZPH)*, which introduces a new set of challenges.

*Equal contribution

Proceedings of the 1st Conference on Topology, Algebra, and Geometry in Data Science(TAG-DS 2025).

While MPH captures rich information, it suffers from the lack of a complete invariant, motivating the search for other informative, though incomplete, invariants Vipond (2020); Corbet et al. (2019); Carrière & Blumberg (2020); Scoccola et al. (2024); Loiseaux et al. (2023); Russold & Kerber (2024); Xin et al. (2023); Mukherjee et al. (2024). The application of ZPH to MPH requires zigzag generalizations of these methods.

In this paper, we find that one such method, GRIL Xin et al. (2023) , which computes a landscape function Bubenik (2015) using generalized ranks of intervals Kim & Mémoli (2021), adapts naturally to the QZPH framework. This adaptation leads to our key contribution, ZZ-GRIL, a new topological invariant that extends the GRIL framework to capture multiscale topological information in time-evolving data. On the theoretical front, we prove the stability of ZZ-GRIL and show that the generalized rank over a specific type of subposet can be computed efficiently using an algorithm from Dey et al. (2024). This allows us to devise a practical algorithm for computing ZZ-GRIL. We demonstrate its value by augmenting machine learning models for tasks like sleep-stage detection from ECG and action classification from multivariate time-series, showing improved performance (Section 5). [1].

A recent paper Flammer & Hüper (2024) considers a similar setup for visualizing data with *spatiotemporal persistence landscapes*. While the setup is similar, our work differs significantly. Notably, Flammer & Hüper (2024) use simple rectangular intervals, allowing a direct modification of a result in Dey et al. (2024). In contrast, we use more general subposets, for which we prove Theorem 4.3, enabling the computation of generalized ranks for zigzag modules and leading to a different overall algorithm. We also provide an efficient algorithm to construct a quasi-zigzag bi-filtration from raw time-varying data and develop a complete ML pipeline that we validate on real-world datasets not considered in Flammer & Hüper (2024).

2 Overview

We begin with essential topological concepts. A *simplicial complex* K is a collection of subsets (simplices) of a finite vertex set V, such that if a simplex σ is in K, then all its subsets are also in K. A (non-zigzag) *filtration* $\mathcal{F}$ is a nested sequence of simplicial complexes indexed by integers: $\mathcal{F} : K_0 \hookrightarrow K_1 \hookrightarrow \cdots \hookrightarrow K_n$. If some inclusions are reversed, we get a *zigzag filtration*, e.g., $\mathcal{Z} : K_0 \hookrightarrow K_1 \hookleftarrow \cdots \hookrightarrow K_n$.

Extending the indexing to two parameters, such as a grid of integers $\mathbb{Z}^2$, we get a bi-filtration. When analyzing time-varying data, a natural construction of a bi-filtration arises where the filtration in one direction is non-zigzag while the filtration in the other direction is a zigzag filtration. Refer to Figure 1 for an example of a moving point cloud data (PCD). A moving PCD with three points in shown in the bottom row. Along the y-direction we have the non-zigzag filtration with increasing distance threshold, while along the x-direction we have a zigzag filtration. We term this mixture a *quasi zigzag bi-filtration*.

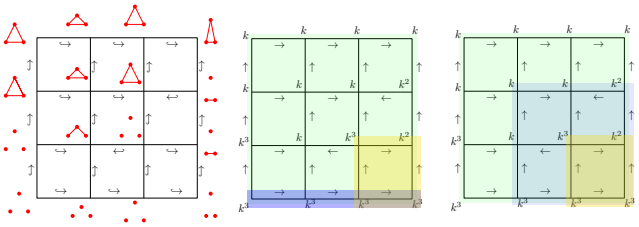

Figure 1: (left) A PCD (bottom row) and a resulting quasi zigzag bi-filtration; (middle) corresponding quasi zigzag persistence module and three intervals (light blue, yellow, green); (right) a rectangle (light yellow) is expanded for obtaining landscape width.

The indexing set of a filtration is a partially ordered set (poset). A non-zigzag filtration can be indexed by the integers $\mathbb{Z}$, while the quasi zigzag bi-filtration is indexed by a product poset $\mathbb{Z}\mathbb{Z} \times \mathbb{Z}$, where $\mathbb{Z}\mathbb{Z}$ denotes a zigzag poset that arises in our case(see Definition 3.1). Applying simplicial homology functor (e.g., H_p) to each complex in the filtration yields a *persistence module*: a collection of vector spaces connected by linear maps according to the inclusions. For a quasi zigzag bi-filtration, this results in a *quasi zigzg persistence module*. In Figure 1 (right), we show the quasi zigzag persistence module for zeroth homology group $H_0(\cdot)$ that represents the number of components.

[1]Code is available at `https://github.com/TDA-Jyamiti/zzgril`

Definition 2.1 (Persistence module)**.** A persistence module over a poset P is a functor $M\colon P \to \mathbf{vec}$, i.e., a collection of vector spaces $\{M_{\mathbf{p}}\}_{\mathbf{p}\in P}$ along with linear maps $M_{\mathbf{p}\to\mathbf{q}}\colon M_{\mathbf{p}} \to M_{\mathbf{q}}$ for every $\mathbf{p} \leq \mathbf{q}$ in P. Here, $\mathbf{vec}$ denotes the category of finite dimensional vector spaces.

Next, we introduce the concept of *intervals*. A finite poset P is connected if for every pair $p, q \in P$ there is a sequence of points $p = r_1, \ldots, r_t = q$ in P so that $r_i \leq r_{i+1}$ or $r_{i+1} \leq r_i$ for all $i \in \{1, \ldots, t-1\}$. An interval $I \subseteq P$ is a connected subposet that is convex w.r.t. poset order, i.e., if $p, q \in I$ and $p \leq r \leq q$, then $r \in I$. Figure 2 depicts an interval in $\mathbb{Z}^2$.

Our framework relies on the *generalized rank* Kim & Mémoli (2021) of a persistence module over a subposet $I \subseteq P$, which can be computed for 2-parameter persistence modules with the algorithm Dey et al. (2024). The generalized rank of the module restricted to I measures the multiplicity of the homological features (# of independent homological classes) that have support over the entire poset I. For example, in Figure 1, the generalized rank, for H_0, over the entire poset (light green) is 1 as one component survives throughout, while the generalized rank over the bottom row (light blue) is 3 and that over the bottom right square (light yellow) is 2.

Definition 2.2 (Generalized rank)**.** Let $M : P \to \mathbf{vec}$ be a persistence module, where P is a finite connected poset. Let $M|_I$ denote the restriction of M to a subposet I of P. Then, the *generalized rank* of M over I is defined as the rank of the canonical linear map from $\mathsf{lim} M|_I$ to $\mathsf{colim} M|_I$

$$\mathsf{rk}^M(I) \coloneqq \mathsf{rank}\left(\mathsf{lim} M|_I \to \mathsf{colim} M|_I\right).$$

We refer the reader to MacLane (1971) for the definitions of limit, colimit, and the construction of the canonical limit-to-colimit map.

A key property of generalized rank is its monotonicity: $\mathsf{rk}^M(I) \leq \mathsf{rk}^M(J)$ for all $J \subseteq I$, where I and J are subposets in P. We use this to define ZZ-GRIL.

To extract the topological information from a quasi zigzag persistence module, we cover the quasi zigzag poset with a specific type of subposets called *worms*. A worm has a notion of a *center* and a *width*. Definition 3.3 gives precise definitions of these terms. We expand each of these worms, i.e., increase the widths while keeping the centers fixed. The monotonicity of the generalized rank ensures that the rank can only decrease by this expansion. In Figure 1(right) a subposet (rectangle) is expanded maximally to decrease the generalized rank from 3 to 1. The width for which the rank drops below a chosen threshold is taken as the ZZ-GRIL value. Defintion 3.4 makes this concept precise. The set of ZZ-GRIL values at the chosen centers makes a vector that captures the topological information in the quasi zigzag persistence module.

3 ZZ-GRIL

In this section, we introduce concepts and definitions required for ZZ-GRIL. Then, we define ZZ-GRIL and discuss its theoretical properties.

Definition 3.1 (Zigzag poset)**.** Zigzag poset $\mathbb{ZZ}$ is defined as a subposet of $\mathbb{Z}^{\mathsf{op}} \times \mathbb{Z}$ given by

$$\mathbb{ZZ} \coloneqq \{(i, j)\colon i \in \mathbb{Z}, j \in \{i, i-1\}\},$$

where $\mathbb{Z}^{\mathsf{op}}$ denotes the opposite poset of $\mathbb{Z}$.

Consider the poset $\mathbb{ZZ} \times \mathbb{Z}$ with the product order, i.e., $(\mathbf{z_1}, z_2) \leq (\mathbf{w_1}, w_2)$ if $\mathbf{z_1} \leq \mathbf{w_1}$ in $\mathbb{ZZ}$ and $z_2 \leq w_2$ in $\mathbb{Z}$. Note that $\mathbb{ZZ} \times \mathbb{Z}$ is equivalent to $\mathbb{Z}^2$ as sets. Thus, every subposet $I \subseteq \mathbb{ZZ} \times \mathbb{Z}$ can be thought as a subposet $I^{\mathbb{Z}^2} \subseteq \mathbb{Z}^2$ that is endowed with $\mathbb{Z}^2$ ordering. Observe that an interval in $\mathbb{Z}^2$ may not remain an interval in $\mathbb{ZZ} \times \mathbb{Z}$ and vice-versa.

We define a *quasi zigzag bi-filtration* as a collection of simplicial complexes $\{K_{\mathbf{p}}\}_{\mathbf{p}\in\mathbb{ZZ}\times\mathbb{Z}}$, where $K_{\mathbf{p}} \subseteq K_{\mathbf{q}}$ for all comparable $\mathbf{p} \leq \mathbf{q}$. Let $\mathbf{vec}$ denote the category of finite dimensional vector spaces.

Definition 3.2 (Quasi zigzag persistence module)**.** A persistence module $M\colon \mathbb{ZZ} \times \mathbb{Z} \to \mathbf{vec}$ is called a *quasi zigzag persistence module*.

We now define a special type of subposet in $\mathbb{ZZ} \times \mathbb{Z}$ which in $\mathbb{Z}^2$ is an ℓ-worm introduced by Xin et al. (2023) for $\ell = 2$.

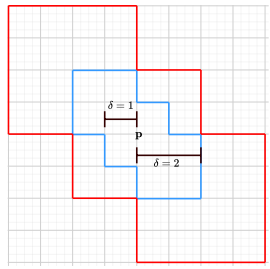

Figure 2: The worm with blue boundary represents a worm centered at $\mathbf{p}$ with width $\delta = 1$. The worm with red boundary represents an expanded worm, centered at $\mathbf{p}$ with width $\delta = 2$.

Definition 3.3 (Worm)**.** Let $\mathbf{p} \in \mathbb{ZZ} \times \mathbb{Z}$ and $\delta \in \mathbb{Z}$ be given. Let $\boxed{\mathbf{p}}_\delta$ denote the δ-square centered at $\mathbf{p}$, i.e., $\boxed{\mathbf{p}}_\delta := \{\mathbf{z} \in \mathbb{ZZ} \times \mathbb{Z} : ||\mathbf{p} - \mathbf{z}||_\infty \leq \delta\}$. Then, a *worm* centered at $\mathbf{p}$ with *width* δ is defined as the union of the two δ-squares $\boxed{\mathbf{q}}_\delta$ centered at points $\mathbf{q} = \mathbf{p} \pm (\delta, -\delta)$ on the off-diagonal line segment along with $\boxed{\mathbf{p}}_\delta$. We denote the worm as $\boxed{\mathbf{p}}^2_\delta$. The superscript denotes the number of δ-squares in the union apart from $\boxed{\mathbf{p}}_\delta$.

We note that computing $|| \cdot ||_\infty$ in $\mathbb{ZZ} \times \mathbb{Z}$ is equivalent to computing it in $\mathbb{Z}^2$ using the notion of set equivalence. We can see that a worm turns out to be an interval in $\mathbb{Z}^2$. Refer to Figure 2 for an illustration of a worm.

Definition 3.4 (ZZ-GRIL)**.** Let M be a quasi zigzag persistence module. Then, the *ZigZag Generalized Rank Invariant Landscape* is a function $\lambda^M : \mathbb{ZZ} \times \mathbb{Z} \times \mathbb{N} \to \mathbb{N}$ defined as

$$\lambda^M(\mathbf{p}, k) := \sup\left\{\delta \geq 0 : \mathsf{rk}^M\left(\boxed{\mathbf{p}}^2_\delta\right) \geq k\right\},$$

where $\mathbf{p} \in \mathbb{ZZ} \times \mathbb{Z}$.

The basic idea of ZZ-GRIL is similar to GRIL. However, we note that the underlying poset structure is very different. For computations, we consider a finite subposet of $\mathbb{ZZ} \times \mathbb{Z}$ and cover it with worms. Then, we compute generalized rank over these worms to define the landscape function (ZZ-GRIL).

3.1 Stability of ZZ-GRIL

We prove the stability of ZZ-GRIL by showing that its perturbation is bounded by the interleaving distance between two quasi zigzag persistence modules. The definition of interleaving distance, $d_{\mathcal{I}}(M, N)$, (see Definition A.3 in Appendix A) between two zigzag persistence modules M, N can be extended to quasi zigzag persistence modules.

There is an alternate notion of proximity on the space of persistence modules which uses Generalized Ranks computed on all intervals. This is known as *erosion distance* Kim & Mémoli (2021). We define a distance similar to erosion distance on the space of quasi zigzag persistence modules based on their generalized ranks computed over worms. For this, we need the notion of ϵ-*thickening*.

Let $\mathbf{I}(\mathbb{ZZ} \times \mathbb{Z})$ denote the collection of all subposets in $\mathbb{ZZ} \times \mathbb{Z}$ such that their corresponding subposets in $\mathbb{Z}^2$ are intervals. For $\epsilon \in \mathbb{Z}_+$, the ϵ-*thickening* of I is defined as

$$I^\epsilon := \{\mathbf{r} \in \mathbb{ZZ} \times \mathbb{Z} : \exists \mathbf{q} \in I \text{ such that } ||\mathbf{r} - \mathbf{q}||_\infty \leq \epsilon\}.$$

Definition 3.5. Let $\mathcal{L}$ denote the collection of all worms in $\mathbb{ZZ} \times \mathbb{Z}$. Let M and N be quasi zigzag persistence modules. The erosion distance is defined as:

$$\begin{aligned} d^{\mathcal{L}}_{\mathcal{E}}(M, N) := \inf_{\epsilon \geq 0}\{\forall \boxed{\mathbf{p}}^2_\delta \in \mathbf{I}(\mathbb{ZZ} \times \mathbb{Z}), \\ \mathsf{rk}^M\left(\boxed{\mathbf{p}}^2_\delta\right) \geq \mathsf{rk}^N\left(\boxed{\mathbf{p}}^2_{\delta+\epsilon}\right) \text{ and} \\ \mathsf{rk}^N\left(\boxed{\mathbf{p}}^2_\delta\right) \geq \mathsf{rk}^M\left(\boxed{\mathbf{p}}^2_{\delta+\epsilon}\right)\}. \end{aligned}$$

Note that $\boxed{\mathbf{p}}^2_{\delta+\epsilon}$ contains the ϵ-thickening of $\boxed{\mathbf{p}}^2_{\delta}$.

Proposition 3.6. *Given two quasi zigzag persistence modules M and N, $d^{\mathcal{L}}_{\mathcal{E}}(M,N) \leq d_{\mathcal{I}}(M,N)$ where $d_{\mathcal{I}}$ denotes the interleaving distance between M and N.*

Proposition 3.6 leads us to Theorem 3.7.

Theorem 3.7. *Let M and N be two quasi zigzag persistence modules. Let λ^M and λ^N denote the* ZZ-GRIL *functions of M and N respectively. Then,*

$$||\lambda^M - \lambda^N||_\infty = d^{\mathcal{L}}_{\mathcal{E}}(M,N) \leq d_{\mathcal{I}}(M,N).$$

We refer the reader to Appendix A for all the proofs.

4 Algorithm

We discuss the details of computing ZZ-GRIL in this section. We begin by proving that generalized rank over a finite subposet in $\mathbb{Z}\mathbb{Z} \times \mathbb{Z}$ can also be computed by computing the number of full bars in the zigzag filtration along the "boundary" of the subposet. To be precise, we introduce the following concepts now. Let P be any finite connected poset. We say a point $q \in P$ *covers* a point $p \in P$, denoted $p \prec q$ or $q \succ p$, if $p \leq q$ and there is no $r \in P$, $r \neq p, q$, so that $p \leq r \leq q$. A *covering* path $p \sim q$ between p and q is either a set of points $\{p = p_0 \prec p_1 \prec \cdots \prec p_\ell = q\}$ or $\{p = p_0 \succ p_1 \succ \cdots \succ p_\ell = q\}$. We define some special points in P. The set of minima in P are the points $P_{min} = \{p \in P \mid \nexists r \neq p \in P \text{ where } r \leq p\}$. Similarly, define the set of maxima P_{max}. For any two points $p, q \in P$, let $p \vee q$ and $p \wedge q$ denote their least upper bound (lub) and largest lower bound (llb) in P if they exist.

Now consider a finite subposet $I \subseteq \mathbb{Z}\mathbb{Z} \times \mathbb{Z}$ whose counterpart $I^{\mathbb{Z}^2}$ with $\mathbb{Z}^2$ ordering becomes an interval. Since $I^{\mathbb{Z}^2} \subset \mathbb{Z}^2$, its points can be sorted with increasing x-coordinates in $\mathbb{Z}^2$. Let $I^{\mathbb{Z}^2}_{min} = \{p_0, \ldots, p_s\}$ and $I^{\mathbb{Z}^2}_{max} = \{q_0, \ldots, q_t\}$ be the set of minima and maxima respectively sorted according to their x-coordinates. The *lower fence* and *upper fence* of $I^{\mathbb{Z}^2}$ are defined as the paths

$$L^{\mathbb{Z}^2}_I := p_0 \sim (p_0 \vee p_1) \sim p_1 \sim (p_1 \vee p_2) \sim p_2 \sim \cdots \sim p_s$$
$$U^{\mathbb{Z}^2}_I := q_0 \sim (q_0 \wedge q_1) \sim q_1 \sim (q_1 \wedge q_2) \sim q_2 \sim \cdots \sim q_t.$$

Consider the boundary $B^{\mathbb{Z}^2}_I$ comprising $L^{\mathbb{Z}^2}_I, U^{\mathbb{Z}^2}_I$, the paths $p_0 \sim q_0$ and $p_s \sim q_t$ going through the top left and bottom right corner points of $I^{\mathbb{Z}^2}$ respectively. Consider the subposet $I \subseteq \mathbb{Z}\mathbb{Z} \times \mathbb{Z}$ and observe that $I_{min}, I_{max} \subseteq B^{\mathbb{Z}^2}_I$ because all other points have one point below and another above in the y-direction. Let $\partial_L I$, $\partial_U I$, and ∂I be minimal paths in $BI^{\mathbb{Z}^2}$ connecting all minima in I_{min}, all maxima in I_{max}, and all minima, maxima together respectively. Drawing an analogy to Dey et al. (2024) we call ∂I a *boundary cap* which is drawn orange in Figure 3.

Next, we appeal to certain results in category theory to claim a result analogous to Dey et al. (2024) that helps computing the generalized ranks with zigzag persistence modules. Treating I as a category with points in I as objects and relations as morphisms, we get $\partial_L I$ and $\partial_U I$ as subcategories. We get a functor $F_L : \partial_L I \to I$ which can be proved to be an *initial functor* (Proposition A.14). Similarly, we also get a *terminal* functor $F_U : I \to \partial_U I$ (initial and terminal functors defined in Definition A.13). The following result connecting initial (terminal) functor to the limit (colimit) is well known, [Chapter 8] Riehl (2014).

Proposition 4.1. *Let $M : I \to \mathbf{vec}$ be a persistence module and $F_L : \partial_L I \to I$ and $F_U : I \to \partial_U I$ be initial and terminal functors. Then, there is an isomorphism $\phi : \lim M \to \lim M|_{\partial_L I}$ from the limit of M to the limit of the restricted module $M|_{\partial_L I}$. Similarly, we have an isomorphism for colimits, $\psi : \mathsf{colim} M|_{\partial_U I} \to \mathsf{colim} M$.*

Proposition 4.2. *$\partial(\partial_L I) = \partial_L I$ and $\partial(\partial_U I) = \partial_U I$.*

Proof. The proof is immediate from the definitions. □

For computations, we bank on the following result which extends Theorem 3.1 in Dey et al. (2024) from 2-parameter persistence modules to the quasi zigzag persistence modules.

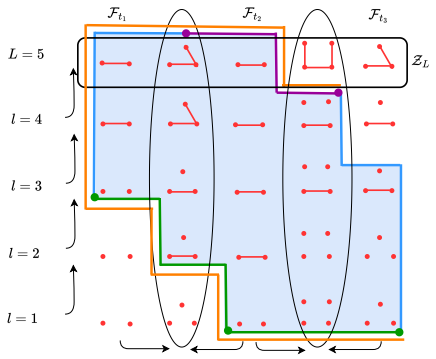

Figure 3: A worm I (shaded) on a quasi zigzag bi-filtration where the direction of the inclusion on horizontal arrows are shown at the bottom and on the vertical arrows on left. The worm centered at (3,3) has width 1. The part of the boundary colored green is $\partial_L I$ connecting the minima shown as green points and the part in purple is $\partial_U I$ connecting the maxima shown as purple points. The *boundary cap* is shown in orange which runs parallel to the boundary for a portion of it. Refer to Figure 5 for the zigzag filtration along the boundary cap of the worm. A sequence of graphs with $T = 3$ time steps is shown with the corresponding graph filtrations: $\mathcal{F}_{t_1}, \mathcal{F}_{t_2}, \mathcal{F}_{t_3}$ each with $L = 5$ levels. The filtration of the unions are encircled by ovals. Zigzag filtration at the topmost level $\mathcal{Z}_L$ is shown in a rectangular box.

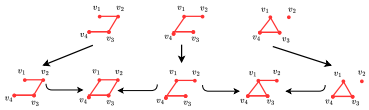

Figure 4: The top row shows a typical input to the ZZ-GRIL framework as a sequence of graphs. This sequence has 3 graphs. Observe that consecutive graphs cannot be linked by an inclusion in either direction because neither is a subgraph of the other. In order to circumvent this problem, we consider the unions of consecutive graphs as an intermediary step, as shown in the bottom row.

A sequence of point clouds (Figure 1) or graphs (Figure 4) or multivariate time-series data (refer section 5 for converting multivariate time-series to a sequence of point clouds or graphs) gives us a collection of simplicial complexes. Every simplex in each simplicial complex is assigned a weight which is derived from the input. We build a quasi zigzag bi-filtration from this collection of simplicial complexes.

Figure 5: This is the zigzag filtration $\mathcal{Z}_{bdry}$ along the boundary cap of the worm shown in Figure 3.

Theorem 4.3. *Let M be a quasi zigzag persistence module and I be a finite interval in the corresponding quasi zigzag poset. Then,* $\mathsf{rk}^M(I) = \mathsf{rk}^M(\partial I)$.

Proof. Let r denote the map $r : \mathsf{lim} I \to \mathsf{colim} I$, and $\phi : \mathsf{lim} I \to \mathsf{lim} \partial_L I$ and $\psi : \mathsf{colim} \partial_U I \to \mathsf{colim} I$ be the isomorphisms guaranteed by Proposition 4.1. Observe that $\mathsf{rk}^M(I) = \mathrm{rank}(\psi^{-1} \circ r \circ \phi^{-1})$. Now let r' denote the map $r' : \mathsf{lim} \partial I \to \mathsf{colim} \partial I$. Consider the isomoprhisms $\phi' : \mathsf{lim} \partial I \to \mathsf{lim} \partial(\partial_L I)$ and $\psi' : \mathsf{colim} \partial(\partial_U I) \to \mathsf{colim} \partial I$ again guaranteed by Proposition 4.1. Then, $\mathsf{rk}^M(\partial I) = \mathrm{rank}(\psi'^{-1} \circ r' \circ \phi'^{-1})$. But, $\psi^{-1} \circ r \circ \phi^{-1} = \psi'^{-1} \circ r' \circ \phi'^{-1}$ because of Proposition 4.2. $\square$

With this background, the pipeline for computing ZZ-GRIL can be described as follows. Suppose that we are given sequential data (sequence of point clouds, sequence of graphs, multivariate time series) with vertex-level correspondences between consecutive time steps. First, we build a quasi zigzag bi-filtration out of this data where time increases in x-direction and the threshold for constructing complexes increases in y-direction. The algorithm for building this bi-filtration out of raw data is described in Appendix B.1 where we make certain choices to make it efficient. Each simplicial complex in the bi-filtration is indexed by a finite grid G in $\mathbb{Z}^2$ because $\mathbb{Z}\mathbb{Z} \times \mathbb{Z}$ is equivalent to $\mathbb{Z}^2$ as sets. We sample a set of center points S from G and compute ZZ-GRIL (Definition 3.4) at each of these center points. Taking advantage of Theorem 4.3, we compute ZZ-GRIL by computing the

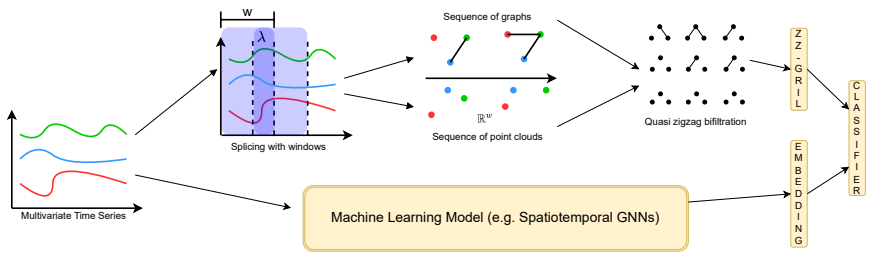

Figure 6: Experimental setup.

zigzag filtration along the boundary cap (shown in Figure 3) of a worm centering each point in S and then computing the number of full bars in the corresponding zigzag persistence module.

4.1 Computing ZZ-GRIL

Given a quasi zigzag bi-filtration, we obtain a quasi zigzag persistence module M by considering the homology $\{H_p(K_{t_i,\alpha_j})\}_{i,j}$ of each simplicial complex. Given a sampled center point $\mathbf{p}$ and the value of rank k, we need to compute $\lambda^M(\mathbf{p}, k)$. Thus, we need to compute the maximal width of the worm such that the value of generalized rank of M over the worm is at least k. We do a binary search over all the possible widths of the worms to arrive at such maximal width for each tuple $(\mathbf{p}, k)$. To compute the generalized rank of M over a worm, we need the zigzag filtration along the boundary cap of the worm, $\mathcal{Z}_{bdry}$. We utilize the stored information about which simplices to add/delete along the horizontal and vertical arrows in order to build $\mathcal{Z}_{bdry}$. Then, we compute the number of full bars in the zigzag persistence module corresponding to $\mathcal{Z}_{bdry}$ using an efficient zigzag algorithm proposed in Dey & Hou (2022) and use it in the binary search routine. The pseudocode of this idea is shown in Algorithm 1 in Appendix B.2. The time complexity of the algorithm is $O(sN^\omega \log T)$, where s is the number of sampled center points, N is the length of the zigzag filtration at the top-most scale level, T is the number of temporal steps and $\omega < 2.37286$ is the matrix multiplication exponent. Refer to Appendix B.3 for the details of time complexity analysis.

5 Experiments

In this section, we report the results of testing ZZ-GRIL on various datasets. We begin by giving a detailed description about the experimental setup. Then, we give a brief description of the datasets, followed by the experimental results. We use the benchmark UEA multivariate time-series Bagnall et al. (2018) datasets to test ZZ-GRIL on multivariate time series data to show that ZZ-GRIL can be applied to datasets from various domains. Further, we test ZZ-GRIL on a targeted application of sleep-stage classification by performing experiments on ISRUC-S3 Khalighi et al. (2016) dataset. In all these experiments, we augment the topological information captured by ZZ-GRIL to one of the specifically tailored machine learning methods on the respective datasets and compare. For each case, we select the machine learning model which has the highest performance to truly test and highlight the value of the topological information added by ZZ-GRIL to an already high-performing specifically tailored model.

5.1 Experimental Setup

Each data instance is a multivariate time-series which we convert into a quasi zigzag bi-filtration. The ZZ-GRIL framework takes in a quasi zigzag bi-filtration as input and provides a topological signature for the sequence as output. We augment the machine learning model with this topological information and train the model for classification. The framework is shown in Figure 6.

Given a sample of multivariate time-series data with m time-series, we splice each time-series into a *sequence of time-series*, each of length w. This splicing is done by a moving window of width w, where consecutive windows have an overlap of λ. We calculate the Pearson correlation coefficient Pearson & Lee (1903) between time-series in each window. We construct a graph for each window, where each time-series (of length w) is a node, and edges with the Pearson correlation values as weights connect them. Thus, we get a complete graph with m nodes. We select the top k

Dataset/Methods	ED-1NN	DTW-1NN-I	DTW-1NN-D	MLSTM-FCN	ShapeNet	WEASEL+MUSE	OS-CNN	MOS-CNN	ZZ-GRIL
FingerMovements	0.550	0.520	0.530	0.580	0.589	0.490	0.568	0.568	**0.590**
Heartbeat	0.620	0.659	0.717	0.663	**0.756**	0.727	0.489	0.604	0.721
MotorImagery	0.510	0.390	0.500	0.510	**0.610**	0.500	0.535	0.515	0.580
NATOPS	0.860	0.850	0.883	0.889	0.883	0.870	**0.968**	0.951	0.850
SelfRegulationSCP2	0.483	0.533	0.539	0.472	**0.578**	0.460	0.532	0.510	0.522

Table 1: Acccuracy comparison of ZZ-GRIL with some existing methods on UEA Multivariate Time Series Classification Datasets. Bold entries denote the best model and gray the second best.

Dataset/Methods	TapNet	TapNet+ZZ-GRIL	TodyNet	TodyNet+ZZ-GRIL
FingerMovements	0.530	0.630	0.570	**0.660**
Heartbeat	0.751	0.751	**0.756**	**0.756**
MotorImagery	0.580	0.600	0.640	**0.660**
NATOPS	0.927	0.922	**0.972**	0.961
SelfRegulationSCP2	0.538	0.544	0.550	**0.600**

Table 2: Test acccuracy of augmenting ZZ-GRIL to TapNet and TodyNet on UEA Multivariate Time Series Classification Datasets.

percentile of these edges to build the final graph in each window. This way we obtain a sequence of graphs. The topological information of this sequence of graphs encodes the evolution of correlation between time series. Alternately, we can also track the evolving time-series by converting it into a sequence of point clouds. From the sequence of time-series described above, we consider each time-series of length w as a point in $\mathbb{R}^w$. Thus, if we have m time series, we have m points in $\mathbb{R}^w$ in each window. Thus, we get a sequence of point clouds in $\mathbb{R}^w$. This sequence of point clouds tracks the evolution of each time-series, and a Vietoris-Rips like construction on this point cloud tracks the evolution of the interaction between time-series. Refer to Figure 6 for an illustration.

5.2 UEA Multivariate Time Series Classification

UEA Multivariate Time Series Classification (MTSC) Bagnall et al. (2018) archive comprises of real-world multivariate time series data and is a widely recognized benchmark in time series analysis. The UEA MTSC collection encompasses a diverse range of application domains such as healthcare (ECG or EEG data), motion recognition (recorded using wearable sensors). See Table 5 in Appendix C.

The datasets are preprocessed and split into training and testing sets. For this set of experiments, we convert the multivariate time series into a sequence of point clouds and compute ZZ-GRIL. We choose 5 datasets which have at least 7 multivariate time series. This ensures that each point cloud, in the sequence of point clouds, has at least 7 points, giving meaningful topological information. Refer to Figure 7 in Appendix C for a visualization of ZZ-GRIL on FingerMovements dataset.

In Table 2, we compare the performance of augmenting ZZ-GRIL to two specifically tailored machine learning models for multivariate time series classification on UEA MTSC datasets, TapNet Zhang et al. (2020) and TodyNet Liu et al. (2024). We can see from the table that ZZ-GRIL is adding meaningful topological information on most datasets which is seen as an improvement in performance. We picked these two models for augmenting ZZ-GRIL because of two reasons: (i) availability and ease-of-use of codebase, (ii) these models already have good performance on UEA MTSC Datasets and we wanted to test the additional value of the topological information that ZZ-GRIL adds. In order to do a comprehensive evaluation of ZZ-GRIL framework, we compare ZZ-GRIL in isolation with other methods Li et al. (2021); Tang et al. (2022); Karim et al. (2019); Schäfer & Leser (2017) on UEA MTSC task. We report the results in Table 1. We see that ZZ-GRIL has comparable performance. Here, we would like to remind the reader that ZZ-GRIL is a general framework which does not need to be tailored for these datasets in particular. Further, to show that we need both the spatial and the temporal information to be captured together, we compare ZZ-GRIL with standard *zigzag persistence* at various spatial scales. We report the results in Table 3. We can see that ZZ-GRIL performs better than standard zigzag persistence even across different scales.

We conduct two ablation studies. First, we find that converting time series into either sequences of point clouds or graphs are both viable approaches for extracting topological information (Appendix C, Table 6). Second, to highlight the value of dynamic topological information, we show that ZZ-GRIL outperforms a "Snapshot PH" baseline, which vectorizes individual non-zigzag persistence modules.

Dataset	Zigzag (scale = 0.5)	Zigzag (scale = 0.7)	Zizag (full graph)	ZZ-GRIL
FingerMovements	0.490	0.490	0.530	**0.590**
Heartbeat	-	0.682	0.687	**0.721**
MotorImagery	0.510	0.520	0.540	**0.580**
NATOPS	-	0.805	0.816	**0.850**
SelfRegulationSCP2	0.483	0.489	0.500	**0.522**

Table 3: Accuracy of 1-parameter zigzag persistence at different scales compared with ZZ-GRIL. We use a Random Forest Classifier for classifying the topological signatures. The empty entries denote that the model does not train on the topological features because the features are not distinct enough at that scale.

Methods	Accuracy	F-1 Overall	F-1 Wake	F-1 N-1	F-1 N2	F-1 N3	F-1 REM
SVM Alickovic & Subasi (2018)	73.3	72.1	86.8	52.3	69.9	78.6	73.1
RF Memar & Faradji (2017)	72.9	70.8	85.8	47.3	70.4	80.9	69.9
MLP+LSTM Dong et al. (2017)	77.9	75.8	86.0	46.9	76.0	87.5	82.8
CNN+BiLSTM Supratak et al. (2017)	78.8	77.9	88.7	60.2	74.6	85.8	80.2
CNN Chambon et al. (2018)	78.1	76.8	87.0	55.0	76.0	85.1	80.9
ARNN+RNN Phan et al. (2019)	78.9	76.3	83.6	43.9	79.3	87.9	86.7
STGCN Jia et al. (2020)	79.9	78.7	87.8	57.4	77.6	86.4	84.1
MSTGCN Jia et al. (2021)	82.1	80.8	**89.4**	59.6	80.6	**89.0**	85.6
STDP-GCN Zhao et al. (2023)	82.6	81.0	83.5	**62.9**	83.1	86.0	90.6
STDP-GCN + ZZ-GRIL	**83.8**	**81.1**	88.6	58.1	**85.4**	82.7	**90.9**

Table 4: Accuracy and F-1 scores of augmenting ZZ-GRIL to STDP-GCN and testing on ISRUC-S3 sleep classification dataset. We report the accuracy and F-1 scores for each model.

As shown in Appendix C Table 7, ZZ-GRIL's superior performance confirms the importance of the features captured by zigzag persistence.

We report the computation times in Table 12 which indicate that ZZ-GRIL is, indeed, practical to use.

5.3 Sleep Stage Classification

We use ISRUC-S3 dataset, which is a part of the ISRUC (Iberian Studies and Research on Sleep) Sleep Dataset Khalighi et al. (2016). ISRUC-S3 contains PSG recordings from 10 subjects. Each recording includes multiple physiological signals, such as: EEG, ECG. The dataset is annotated with sleep stage labels for each epoch (30 second window). There are 5 sleep stage labels: Wake (W), N1, N2, N3 (non-REM stages) and REM. We use STDP-GCN Zhao et al. (2023) as the machine learning model to augment. In this experiment, we convert the time series into sequence of graphs and compare the performance in Table 4. We can see that augmenting ZZ-GRIL increases both the accuracy and the overall F-1 score.

We would like to clarify that the aim, for both sets of experiments, is primarily to show that an increase in accuracy upon augmentation signifies that ZZ-GRIL captures meaningful topological information which can be used to improve the existing models.

6 Conclusion

In this paper, we proposed QZPH as a framework to capture both static and dynamic topological features in time-varying data. We proposed ZZ-GRIL, a stable and computationally efficient topological invariant to address the challenges of integrating MPH and ZPH. Through applications in various domains, including sleep-stage detection, we showed that augmenting machine learning models with ZZ-GRIL improves the performance. These results highlight the potential of integrating topological information to address complex challenges while analyzing time-evolving data.

7 Acknowledgement

This work is partially supported by NSF grants DMS-2301360 and CCF-2437030. We acknowledge the discussion with Michael Lesnick who pointed out the theory about initial/terminal functors in the context of limits and colimits.

References

Alickovic, E. and Subasi, A. Ensemble SVM method for automatic sleep stage classification. *IEEE Transactions on Instrumentation and Measurement*, 67(6):1258–1265, 2018. URL https://ieeexplore.ieee.org/document/8292946.

Bagnall, A., Dau, H. A., Lines, J., Flynn, M., Large, J., Bostrom, A., Southam, P., and Keogh, E. The UEA multivariate time series classification archive, 2018, 2018. URL https://arxiv.org/abs/1811.00075.

Beltramo, G., Skraba, P., Andreeva, R., Sarkar, R., Giarratano, Y., and Bernabeu, M. O. Euler characteristic surfaces. *Foundations of Data Science*, 4(4):505–536, 2022. URL https://www.aimsciences.org/article/doi/10.3934/fods.2021027?viewType=html.

Botnan, M. and Lesnick, M. Algebraic stability of zigzag persistence modules. *Algebraic & geometric topology*, 18(6):3133–3204, 2018. URL https://msp.org/agt/2018/18-6/agt-v18-n6-p01-s.pdf.

Botnan, M. B. and Lesnick, M. An introduction to multiparameter persistence, 2023. https://arxiv.org/abs/2203.14289.

Box, G. and Jenkins, G. M. *Time Series Analysis: Forecasting and Control*. Holden-Day, 1976.

Bubenik, P. 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.

Carlsson, G. and de Silva, V. Zigzag persistence. *Foundations of computational mathematics*, 10: 367–405, 2010.

Carrière, M. and Blumberg, A. Multiparameter persistence image for topological machine learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (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.

Chambon, S., Galtier, M. N., Arnal, P. J., Wainrib, G., and Gramfort, A. A deep learning architecture for temporal sleep stage classification using multivariate and multimodal time series. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 26(4):758–769, 2018. URL https://ieeexplore.ieee.org/document/8307462.

Chen, Y., Segovia, I., and Gel, Y. R. Z-GCNETs: Time zigzags at graph convolutional networks for time series forecasting. In *International Conference on Machine Learning*, pp. 1684–1694. PMLR, 2021. URL https://proceedings.mlr.press/v139/chen21o/chen21o.pdf.

Chu, P., Wang, J., You, Q., Ling, H., and Liu, Z. Transmot: Spatial-temporal graph transformer for multiple object tracking. In *Proceedings of the IEEE/CVF Winter Conference on applications of computer vision*, pp. 4870–4880, 2023. URL https://openaccess.thecvf.com/content/WACV2023/papers/Chu_TransMOT_Spatial-Temporal_Graph_Transformer_for_Multiple_Object_Tracking_WACV_2023_paper.pdf.

Corbet, R., Fugacci, U., Kerber, M., Landi, C., and Wang, B. A kernel for multi-parameter 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.

Coskunuzer, B., Segovia-Dominguez, I., Chen, Y., and Gel, Y. R. Time-aware knowledge representations of dynamic objects with multidimensional persistence. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 11678–11686, 2024. URL https://ojs.aaai.org/index.php/AAAI/article/view/29051/29991.

Dey, T. K. and Hou, T. Fast Computation of Zigzag Persistence. In Chechik, S., Navarro, G., Rotenberg, E., and Herman, G. (eds.), *30th Annual European Symposium on Algorithms (ESA 2022)*, volume 244 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 43:1–43:15, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-247-1. doi: 10.4230/LIPIcs.ESA.2022.43. URL https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ESA.2022.43.

Dey, T. K. and Wang, Y. *Computational Topology for Data Analysis*. Cambridge University Press, 2022. doi: 10.1017/9781009099950. URL https://www.cs.purdue.edu/homes/tamaldey/book/CTDAbook/CTDAbook.pdf.

Dey, T. K., Kim, W., and Mémoli, F. Computing generalized rank invariant for 2-parameter persistence modules via zigzag persistence and its applications. *Discret. Comput. Geom.*, 71 (1):67–94, 2024. doi: 10.1007/S00454-023-00584-Z. URL https://doi.org/10.1007/s00454-023-00584-z.

Dong, H., Supratak, A., Pan, W., Wu, C., Matthews, P. M., and Guo, Y. Mixed neural network approach for temporal sleep stage classification. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 26(2):324–333, 2017. URL https://ieeexplore.ieee.org/document/7995122.

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

Fan, H., Yu, X., Ding, Y., Yang, Y., and Kankanhalli, M. PSTNet: Point spatio-temporal convolution on point cloud sequences. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=O3bqkf_Puys.

Flammer, M. and Hüper, K. Spatiotemporal persistence landscapes, 2024. URL https://arxiv.org/abs/2412.11925.

Hacquard, O. and Lebovici, V. Euler characteristic tools for topological data analysis. *Journal of Machine Learning Research*, 25(240):1–39, 2024. URL http://jmlr.org/papers/v25/23-0353.html.

Hamilton, J. D. *Time Series Analysis*. Princeton University Press, 1994. URL https://www.worldcat.org/title/time-series-analysis/oclc/1194970663&referer=brief_results.

Huang, S., Xie, Y., Zhu, S.-C., and Zhu, Y. Spatio-temporal self-supervised representation learning for 3d point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6535–6545, 2021. URL https://openaccess.thecvf.com/content/ICCV2021/papers/Huang_Spatio-Temporal_Self-Supervised_Representation_Learning_for_3D_Point_Clouds_ICCV_2021_paper.pdf.

Jia, Z., Lin, Y., Wang, J., Zhou, R., Ning, X., He, Y., and Zhao, Y. GraphSleepNet: Adaptive spatial-temporal graph convolutional networks for sleep stage classification. In *IJCAI*, volume 2021, pp. 1324–1330, 2020. URL https://www.ijcai.org/proceedings/2020/0184.pdf.

Jia, Z., Lin, Y., Wang, J., Ning, X., He, Y., Zhou, R., Zhou, Y., and Li-wei, H. L. Multi-view spatial-temporal graph convolutional networks with domain generalization for sleep stage classification. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 29:1977–1986, 2021. URL https://ieeexplore.ieee.org/abstract/document/9530406.

Kan, J., Hu, K., Hagenbuchner, M., Tsoi, A. C., Bennamoun, M., and Wang, Z. Sign language translation with hierarchical spatio-temporal graph neural network. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 3367–3376, 2022. URL https://openaccess.thecvf.com/content/WACV2022/papers/Kan_Sign_Language_Translation_With_Hierarchical_Spatio-Temporal_Graph_Neural_Network_WACV_2022_paper.pdf.

Karim, F., Majumdar, S., Darabi, H., and Harford, S. Multivariate LSTM-FCNs for time series classification. *Neural networks*, 116:237–245, 2019. URL https://www.sciencedirect.com/science/article/pii/S0893608019301200?casa_token=PDVs4qbGRbIAAAAA:zy34UVvHn2z55zyu_P3ulILMj82TQBxrTtOLrBAQiVh_aVks7unsEXFEqnq2ywsCkcVDLRvI1Q.

Khalighi, S., Sousa, T., Santos, J. M., and Nunes, U. ISRUC-Sleep: A comprehensive public dataset for sleep researchers. *Computer Methods and Programs in Biomedicine*, 124:180–192, 2016. ISSN 0169-2607. doi: https://doi.org/10.1016/j.cmpb.2015.10.013. URL https://www.sciencedirect.com/science/article/pii/S0169260715002734.

Kim, W. and Mémoli, F. Generalized persistence diagrams for persistence modules over posets. *Journal of Applied and Computational Topology*, 5(4):533–581, 12 2021. ISSN 2367-1734. doi: 10.1007/s41468-021-00075-1. URL https://doi.org/10.1007/s41468-021-00075-1.

Li, G., Choi, B., Xu, J., Bhowmick, S. S., Chun, K.-P., and Wong, G. L.-H. Shapenet: A shapelet-neural network approach for multivariate time series classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 8375–8383, 2021.

Liu, H., Yang, D., Liu, X., Chen, X., Liang, Z., Wang, H., Cui, Y., and Gu, J. TodyNet: Temporal dynamic graph neural network for multivariate time series classification. *Information Sciences*, 677:120914, 2024. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2024.120914. URL https://www.sciencedirect.com/science/article/pii/S0020025524008284.

Liu, X., Yan, M., and Bohg, J. Meteornet: Deep learning on dynamic 3d point cloud sequences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9246–9255, 2019. URL https://openaccess.thecvf.com/content_ICCV_2019/papers/Liu_MeteorNet_Deep_Learning_on_Dynamic_3D_Point_Cloud_Sequences_ICCV_2019_paper.pdf.

Loiseaux, D., Scoccola, L., Carrière, M., Botnan, M. B., and Oudot, S. Stable vectorization of multiparameter persistent homology using signed barcodes as measures. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/pdf?id=4mwORQjAim.

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

Memar, P. and Faradji, F. A novel multi-class EEG-based sleep stage classification system. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 26(1):84–95, 2017. URL https://ieeexplore.ieee.org/document/8116601.

Mukherjee, S., Samaga, S. N., Xin, C., Oudot, S., and Dey, T. K. D-GRIL: End-to-end topological learning with 2-parameter persistence, 2024. URL https://arxiv.org/abs/2406.07100.

Myers, A., Muñoz, D., Khasawneh, F. A., and Munch, E. Temporal network analysis using zigzag persistence. *EPJ Data Science*, 12(1):6, 2023. URL https://epjdatascience.springeropen.com/articles/10.1140/epjds/s13688-023-00379-5.

Oreshkin, B. N., Amini, A., Coyle, L., and Coates, M. FC-GAGA: Fully connected gated graph architecture for spatio-temporal traffic forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 9233–9241, 2021. URL https://ojs.aaai.org/index.php/AAAI/article/view/17114/16921.

Pearson, K. and Lee, A. On the laws of inheritance in man: I. inheritance of physical characters. *Biometrika*, 2(4):357–462, 11 1903. ISSN 0006-3444. doi: 10.1093/biomet/2.4.357. URL https://doi.org/10.1093/biomet/2.4.357.

Phan, H., Andreotti, F., Cooray, N., Chén, O. Y., and De Vos, M. SeqSleepNet: end-to-end hierarchical recurrent neural network for sequence-to-sequence automatic sleep staging. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 27(3):400–410, 2019. URL https://ieeexplore.ieee.org/document/8631195.

Rempe, D., Birdal, T., Zhao, Y., Gojcic, Z., Sridhar, S., and Guibas, L. J. Caspr: Learning canonical spatiotemporal point cloud representations. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 13688–13701. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/9de6d14fff9806d4bcd1ef555be766cd-Paper.pdf.

Riehl, E. *Categorical homotopy theory*. New Mathematical Monographs, Series Number 24. Cambridge University Press, 2014.

Russold, F. and Kerber, M. Graphcode: Learning from multiparameter persistent homology using graph neural networks. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/4822d9adc9cec7a39e254d007aa78276-Abstract-Conference.html.

Schäfer, P. and Leser, U. Multivariate time series classification with WEASEL+MUSE. *arXiv preprint arXiv:1711.11343*, 2017. URL https://arxiv.org/pdf/1711.11343.

Scoccola, L., Setlur, S., Loiseaux, D., Carrière, M., and Oudot, S. Differentiability and optimization of multiparameter persistent homology. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 43986–44011. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/scoccola24a.html.

Supratak, A., Dong, H., Wu, C., and Guo, Y. DeepSleepNet: A model for automatic sleep stage scoring based on raw single-channel EEG. *IEEE transactions on neural systems and rehabilitation engineering*, 25(11):1998–2008, 2017. URL https://ieeexplore.ieee.org/document/7961240.

Tang, W., Long, G., Liu, L., Zhou, T., Blumenstein, M., and Jiang, J. Omni-scale CNNs: a simple and effective kernel size configuration for time series classification. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=PDYs7Z2XFGv.

Tinarrage, R., Ponciano, J. R., Linhares, C. D. G., Traina, A. J. M., and Poco, J. ZigzagNetVis: Suggesting temporal resolutions for graph visualization using zigzag persistence. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–18, 2025. doi: 10.1109/TVCG.2025.3528197. URL https://www.computer.org/csdl/journal/tg/5555/01/10844578/23zUk2JOsr6.

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

Wu, Z., Pan, S., Long, G., Jiang, J., Chang, X., and Zhang, C. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 753–763, 2020. URL https://dl.acm.org/doi/pdf/10.1145/3394486.3403118.

Xin, C., Mukherjee, S., Samaga, S. N., and Dey, T. K. GRIL: A 2-parameter Persistence Based Vectorization for Machine Learning. In *Proceedings of 2nd Annual Workshop on Topology, Algebra, and Geometry in Machine Learning (TAG-ML)*, volume 221 of *Proceedings of Machine Learning Research*, pp. 313–333. PMLR, 7 2023. URL https://proceedings.mlr.press/v221/xin23a.html.

Yu, B., Yin, H., and Zhu, Z. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pp. 3634–3640, 2018. URL https://www.ijcai.org/proceedings/2018/0505.pdf.

Zhang, X., Gao, Y., Lin, J., and Lu, C.-T. Tapnet: Multivariate time series classification with attentional prototypical network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 6845–6852, 2020. URL https://ojs.aaai.org/index.php/AAAI/article/view/6165.

Zhao, Y., Lin, X., Zhang, Z., Wang, X., He, X., and Yang, L. STDP-based adaptive graph convolutional networks for automatic sleep staging. *Frontiers in Neuroscience*, 17, 2023. ISSN 1662-453X. doi: 10.3389/fnins.2023.1158246. URL https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2023.1158246.

A Formal definitions and proofs

There is a notion of proximity on the space of zigzag modules in terms of the *interleaving distance*. In Botnan & Lesnick (2018), the authors define interleaving distance between two zigzag modules by including them into $\mathbb{R}^{\mathsf{op}} \times \mathbb{R}$-indexed modules.

The interleaving distance on $\mathbb{R}^n$-indexed persistence modules is known and well-defined. We briefly recall the definition here. We refer the readers to Botnan & Lesnick (2018) for additional details.

Definition A.1 (u-shift functor)**.** The u-shift functor $(-)_u \colon \mathbf{vec}^{\mathbb{R}^n} \to \mathbf{vec}^{\mathbb{R}^n}$, for $u \in \mathbb{R}^n$, is defined as follows:

1. For $M \in \mathbf{vec}^{\mathbb{R}^n}$, M_u is defined as $M_u(x) = M(x+u)$ for all $x \in \mathbb{R}^n$ and $M_u(x_1 \leq x_2) = M(x_1 + u \leq x_2 + u)$ for all $x_1 \leq x_2 \in \mathbb{R}^n$,
2. Let $M, N \in \mathbf{vec}^{\mathbb{R}^n}$. Let $F \colon M \to N$ be a morphism. Then, the corresponding morphism $F_u \colon M_u \to N_u$ is defnied as $F_u(x) = F(x+u) \colon M_u(x) \to N_u(x)$ for all $x \in \mathbb{R}^n$.

Definition A.2 (ϵ-interleaving)**.** Let $M, N \in \mathbf{vec}^{\mathbb{R}^n}$. Let $\epsilon \in [0, \infty)$ be given. We will denote $(-)_{\boldsymbol{\epsilon}}$ to be the shift functor corresponding to the vector $\boldsymbol{\epsilon} = \epsilon(1, 1, \ldots, 1)$. We say M and N are ϵ-interleaved if there are natural transformations $F \colon M \to N_{\boldsymbol{\epsilon}}$ and $G \colon N \to M_{\boldsymbol{\epsilon}}$ such that

1. $G_{\boldsymbol{\epsilon}} \circ F = \varphi_M^{2\boldsymbol{\epsilon}}$,
2. $F_{\boldsymbol{\epsilon}} \circ G = \varphi_N^{2\boldsymbol{\epsilon}}$,

where $\varphi_M^u \colon M \to M_u$ is the natural transformation whose restriction to each $M(x)$ is the linear map $M(x \leq x + u)$ for all $x \in \mathbb{R}^n$.

Definition A.3 (Interleaving distance)**.** Let $M, N \in \mathbf{vec}^{\mathbb{R}^n}$. The interleaving distance $d_{\mathcal{I}}(M, N)$ between M and N is defined as

$$d_{\mathcal{I}}(M, N) \coloneqq \inf\{\epsilon \geq 0 \colon M \text{ and } N \text{ are } \epsilon\text{-interleaved}\}$$

and $d_{\mathcal{I}}(M, N) = \infty$ if there exists no interleaving.

We need the following two definitions to define interleaving distance between two zigzag modules.

Definition A.4 (Left Kan Extension)**.** Let P and Q be two posets. Let $F \colon P \to Q$ be a functor. Let $P[F \leq q]$ denote the set $P[F \leq q] \coloneqq \{p \in P \colon F(p) \leq q\}$. Given a persistence module $M \colon P \to \mathbf{vec}$, the *left Kan extension* of M along F is a functor $\mathrm{Lan}_F(M) \colon Q \to \mathbf{vec}$ given by

$$\mathrm{Lan}_F(M)(q) \coloneqq \mathsf{colim} M|_{P[F \leq q]},$$

along with internal morphisms given by the universality of colimits.

Definition A.5 (Block Extension Functor Botnan & Lesnick (2018))**.** Let $\mathbb{U} \subset \mathbb{R}^{\mathsf{op}} \times \mathbb{R}$ denote the poset $\mathbb{U} \coloneqq \{(a, b) \colon a \leq b\}$. Let $\mathfrak{i} \colon \mathbb{ZZ} \to \mathbb{R}^{\mathsf{op}} \times \mathbb{R}$ denote the inclusion. Let $(-)|_{\mathbb{U}} \colon \mathbf{vec}^{\mathbb{R}^{\mathsf{op}} \times \mathbb{R}} \to \mathbf{vec}^{\mathbb{U}}$ denote the restriction. Then, the *block extension functor* $E \colon \mathbf{vec}^{\mathbb{ZZ}} \to \mathbf{vec}^{\mathbb{U}}$ is defined as

$$E \coloneqq (-)|_{\mathbb{U}} \circ \mathrm{Lan}_{\mathfrak{i}}(\circ).$$

Definition A.6 (Interleaving distance on zigzag modules)**.** Let M, N be two zigzag modules. Then

$$d_{\mathcal{I}}(M, N) \coloneqq d_{\mathcal{I}}(E(M), E(N)).$$

We extend this definition of interleaving distance to quasi zigzag persistence modules. Let $\mathbb{UU} \subset \mathbb{R}^{\mathsf{op}} \times \mathbb{R} \times \mathbb{R}$ be the analog of $\mathbb{U}$, i.e, $\mathbb{UU} \coloneqq \{(a, b, c) \colon a \leq b\}$. Let $\tilde{\mathfrak{i}} \colon \mathbb{ZZ} \times \mathbb{Z} \to \mathbb{R}^{\mathsf{op}} \times \mathbb{R} \times \mathbb{R}$ denote the inclusion. Then, define $\tilde{E} \coloneqq (-)|_{\mathbb{UU}} \circ \mathrm{Lan}_{\tilde{\mathfrak{i}}}(\circ)$, analogous to the block extension functor.

Definition A.7. Let M, N be two quasi zigzag persistence modules. Then, the interleaving distance between M and N is given by

$$d_{\mathcal{I}}(M, N) \coloneqq d_{\mathcal{I}}(\tilde{E}(M), \tilde{E}(N)).$$

Lemma A.8. *$\tilde{E}$ sends interval modules on $\mathbb{ZZ} \times \mathbb{Z}$ to block interval modules.*

This is analogous to Lemma 4.1 in Botnan & Lesnick (2018).

Lemma A.9. *Let M be a quasi zigzag persistence module. If $M \cong \bigoplus_{k\in K} I_k$, then $\tilde{E}(M) \cong \bigoplus_{k\in K} \tilde{E}(I_k)$, where K is an indexing set.*

Proof. The functor $\mathrm{Lan}_{\tilde{\mathrm{i}}}$ preserves direct sums MacLane (1971), as it is left-adjoint to the canonical restriction functor. The restriction functor also preserves direct sums. These two facts combined with Lemma A.8 give the result. □

Definition A.10. Let $\mathbf{I}(\mathbb{Z}\mathbb{Z} \times \mathbb{Z})$ denote the collection of all subposets in $\mathbb{Z}\mathbb{Z} \times \mathbb{Z}$ such that their corresponding subposets in $\mathbb{Z}^2$ are intervals. Let M and N be two quasi zigzag persistence modules. The erosion distance is defined as:

$$\begin{aligned} d_{\mathcal{E}}(M,N) \coloneqq \inf_{\epsilon\geq 0}\{\forall I \in \mathbf{I}(\mathbb{Z}\mathbb{Z}\times\mathbb{Z}), \\ \mathsf{rk}^M(I) \geq \mathsf{rk}^N(I^\epsilon) \text{ and} \\ \mathsf{rk}^N(I) \geq \mathsf{rk}^M(I^\epsilon)\} \end{aligned}$$

Definition A.11. Let $\mathcal{L}$ denote the collection of all worms in $\mathbb{Z}\mathbb{Z} \times \mathbb{Z}$. Let M and N be quasi zigzag persistence modules. The erosion distance can be defined as:

$$\begin{aligned} d^{\mathcal{L}}_{\mathcal{E}}(M,N) \coloneqq \inf_{\epsilon\geq 0}\{\forall \boxed{\mathbf{p}}^2_\delta \in \mathbf{I}(\mathbb{Z}\mathbb{Z}\times\mathbb{Z}), \\ \mathsf{rk}^M\left(\boxed{\mathbf{p}}^2_\delta\right) \geq \mathsf{rk}^N\left(\boxed{\mathbf{p}}^2_{\delta+\epsilon}\right) \text{ and} \\ \mathsf{rk}^N\left(\boxed{\mathbf{p}}^2_\delta\right) \geq \mathsf{rk}^M\left(\boxed{\mathbf{p}}^2_{\delta+\epsilon}\right)\}. \end{aligned}$$

***Proposition** 3.6.* Given two quasi zigzag persistence modules M and N, $d^{\mathcal{L}}_{\mathcal{E}}(M,N) \leq d_{\mathcal{I}}(M,N)$ where $d_{\mathcal{I}}$ denotes the interleaving distance between M and N.

Proof. First, it is obvious that $d^{\mathcal{L}}_{\mathcal{E}}(M,N) \leq d_{\mathcal{E}}(M,N)$. Now, $d_{\mathcal{I}}(M,N) \coloneqq d_{\mathcal{I}}(\tilde{E}(M),\tilde{E}(N))$. Let I be a subposet in $\mathbb{Z}\mathbb{Z}\times\mathbb{Z}$ such that its corresponding subposet in $\mathbb{Z}^2$ is an interval. By Lemma A.8 and Lemma A.9, and the fact that generalized rank over a given subposet counts the number of intervals that contain the given subposet, we get $\mathsf{rk}^M(I) = \mathsf{rk}^{\tilde{E}(M)}(\tilde{\mathrm{i}}(I) \cap \mathbb{U}\mathbb{U})$. This gives us $d_{\mathcal{E}}(M,N) = d_{\mathcal{E}}(\tilde{E}(M),\tilde{E}(N))$. In Kim & Mémoli (2021), the authors show that $d_{\mathcal{E}}(V,W) \leq d_{\mathcal{I}}(V,W)$, where $V,W\colon \mathbb{R}^n \to \mathbf{vec}$ are $\mathbb{R}^n$-indexed persistence modules. Thus, we get $d^{\mathcal{L}}_{\mathcal{E}}(M,N) \leq d_{\mathcal{E}}(M,N) = d_{\mathcal{E}}(\tilde{E}(M),\tilde{E}(N)) \leq d_{\mathcal{I}}(\tilde{E}(M),\tilde{E}(N)) = d_{\mathcal{I}}(M,N)$.

□

***Theorem** 3.7.* Let M and N be two quasi zigzag persistence modules. Let λ^M and λ^N denote the ZZ-GRIL functions of M and N respectively. Then,

$$||\lambda^M - \lambda^N||_\infty = d^{\mathcal{L}}_{\mathcal{E}}(M,N) \leq d_{\mathcal{I}}(M,N).$$

Proof. We show that $||\lambda^M - \lambda^N||_\infty = d^{\mathcal{L}}_{\mathcal{E}}(M,N)$.

To see $||\lambda^M - \lambda^N||_\infty \leq d^{\mathcal{L}}_{\mathcal{E}}(M,N)$, fix $\mathbf{p}, k$, and let $\lambda^M(\mathbf{p},k) = \delta_M$ and $\lambda^N(\mathbf{p},k) = \delta_N$. WLOG, assume $\delta_M \geq \delta_N$. Let $d^{\mathcal{L}}_{\mathcal{E}}(M,N) = \epsilon$. Therefore, by definition of $d^{\mathcal{L}}_{\mathcal{E}}$, we have $\mathsf{rk}^M\left(\boxed{\mathbf{p}}^2_{\delta_M}\right) = k$ and $\mathsf{rk}^M\left(\boxed{\mathbf{p}}^2_{\delta_N+\epsilon}\right) \leq \mathsf{rk}^N\left(\boxed{\mathbf{p}}^2_{\delta_N}\right) = k$. Thus, by the definition of ZZ-GRIL, we get $\delta_N + \epsilon \geq \delta_M$, i.e., $\delta_M - \delta_N \leq \epsilon = d^{\mathcal{L}}_{\mathcal{E}}$.

To see $||\lambda^M - \lambda^N||_\infty \geq d^{\mathcal{L}}_{\mathcal{E}}(M,N)$, fix $\mathbf{p}, k$, and let $|\lambda^M(\mathbf{p},k) - \lambda^N(\mathbf{p},k)| = \epsilon$. Let $\boxed{\mathbf{p}}^2_\delta$ be a worm. Let $k = \mathsf{rk}^N\left(\boxed{\mathbf{p}}^2_{\delta+\epsilon}\right)$. Then, $\lambda^N(\mathbf{p},k) \geq \delta + \epsilon$. Thus, $|\lambda^M(\mathbf{p},k) - \lambda^N(\mathbf{p},k)| = \epsilon$ and $\lambda^N(\mathbf{p},k) \geq \delta + \epsilon$ give $\lambda^M(\mathbf{p},k) \geq d$. Therefore, $\mathsf{rk}^M\left(\boxed{\mathbf{p}}^2_\delta\right) \geq k = \mathsf{rk}^N\left(\boxed{\mathbf{p}}^2_{\delta+\epsilon}\right)$. Similarly, we can show $\mathsf{rk}^N\left(\boxed{\mathbf{p}}^2_\delta\right) \geq \mathsf{rk}^M\left(\boxed{\mathbf{p}}^2_{\delta+\epsilon}\right)$. Thus, by definition of $d^{\mathcal{L}}_{\mathcal{E}}$, we get that $||\lambda^M - \lambda^N||_\infty \geq d^{\mathcal{L}}_{\mathcal{E}}(M,N)$.

□

Here, we define initial functors and terminal functors that we use in Section 4. Further, we prove Proposition A.14 which is used in Proposition 4.1 leading to the final result in Theorem 4.3.

Definition A.12. A connected subposet $P' \subseteq P$ is called *initial* if for every $p \in P \setminus P'$, the downset $\downarrow p = \{q \in P \mid q \neq p \,\&\, q \leq p\}$ intersects P' in a connected poset. Similalrly, P' is called *terminal* if for every $p \in P \setminus P'$, the upset $\uparrow p = \{q \in P \mid q \neq p \,\&\, q \geq p\}$ intersects P' in a connected poset.

Definition A.13. We call a functor $F : P' \to P$ for $P' \subseteq P$ *initial* if P' is initial. Similarly, we call $F : P \to P'$ *terminal* if P' is terminal.

Let $I, \partial_L I$ and $\partial_U I$ be as defined in Section 4. Treating I as a category with points in I as objects and relations as morphisms, we get $\partial_L U$ and $\partial_U I$ as subcategories. Then, we get functors $F_L \colon \partial_L I \to I$ induced by inclusion and $F_U \colon I \to \partial_U I$ induced by projection.

Proposition A.14. *$F_L : \partial_L I \to I$ is an initial functor and $F_U : I \to \partial_U I$ is a terminal functor.*

Proof. It is sufficient to prove that $\partial_L I$ is initial for F_L and $\partial_U I$ is terminal for F_U. We only show that $\partial_L I$ is initial and the proof for $\partial_U I$ being terminal is similar.

Let $p \in I \setminus \partial_L I$ be any point. Let p^- and p^+ be two points (if exist) where $p^- \prec p$ and $p^+ \succ p$ in $\mathbb{Z}^2$ with $p^-_x < p_x < p^+_x$. By definition of the quasi zigzag poset, either $p^- > p$ and $p^+ > p$, or $p^- < p$ and $p^+ < p$ in the poset $\mathbb{Z}\mathbb{Z} \times \mathbb{Z}$.

In the first case, the downset $\downarrow p$ consists of all points $p' \neq p$ that are vertically below p, that is, $p'_x = p_x$ and $p'_y < p_y$. This means $\downarrow p$ intersects $\partial_L I$ in a connected poset $p'_0 < \cdots < p'_k$ where each p'_i has the same x-coordinate as p.

In the second case, the downset $\downarrow p$ consists of three sets of points Y^{p^-}, Y^p, and Y^{p^+} that are vertically below p^-, p, and p^+ respectively. Each of Y^{p^-}, Y^p and Y^{p^+} intersects $\partial_L I$ in a connected poset. They can be disconnected as a whole only if there is a point $q \in \partial_L I$ so that $p^-_x < q_x < p_x$ or $p_x < q_x < p^+_x$. But, neither is possible because p^- and p^+ are points in I covered by p. $\square$

B Algorithmic Details

B.1 Building Quasi Zigzag Bi-filtration

We explain the algorithm by considering an example of sequential graph data, as shown in Figure 4. We assume that each graph has edge-weights. In the top row of Figure 4, notice that the first two graphs can not be linked by an inclusion in any direction, i.e., neither is a subgraph of the other. This is because, there is an inclusion of an edge (v_2, v_3) as well as a deletion (v_1, v_4). We circumvent this problem by clubbing all inclusions together followed by all deletions. This is equivalent to considering the union of the two graphs as the intermediary step and then deleting the edges which are not present in the previous graph. Refer to the bottom row of Figure 4, where the intermediary union graphs are shown along with the original graphs.

A sequence of T number of graphs is converted into a zigzag filtration $\mathcal{Z}_L$ of length $2T - 1$ by the above procedure because every consecutive pair of graphs introduces a union in between. For each graph in the zigzag filtration thus obtained, we construct its *graph filtration* based on edge-weights; see e.g. Dey & Wang (2022); Edelsbrunner & Harer (2010). We ensure that the length of each graph filtration remains the same by considering the sublevel sets of exactly L levels. The zigzag filtration at the highest level, $\mathcal{Z}_L$, can be pulled back to each lower level l. This ensures that we get a zigzag filtration $\mathcal{Z}_l$ at each level $1 \leq l < L$ of the filtration. This gives us a quasi zigzag bi-filtration. Refer to Figure 3 for an illustation of a quasi zigzag bi-filtration.

While implementing this procedure, we need not compute the union graphs explicitly saving both storage and time because all necessary information is already present in the corresponding component graphs. We use the following procedure to efficiently build the quasi zigzag bi-filtration.

Construct the graph filtration $\mathcal{F}_{t_i}$ of each of the T graphs in the sequential graph data, where $1 \leq t_i \leq T$. Let σ be a simplex that is inserted at the level l in the filtration $\mathcal{F}_{t_i}$. We have three possible scenarios for σ in $\mathcal{F}_{t_{i+1}}$:

Case (i) σ is inserted in $\mathcal{F}_{t_{i+1}}$ at m for some $m < l$: In this case, σ needs to be added on all the horizontal inclusion arrows $K_{t_i,w} \hookrightarrow K_{t_i,w} \cup K_{t_{i+1},w}$ for $m \leqslant w < l$.

Case (ii) σ is inserted in $\mathcal{F}_{t_{i+1}}$ at m for some $m \geqslant l$: In this case, σ needs to be added on all the horizontal inclusion arrows $K_{t_{i+1},w} \hookrightarrow K_{t_i,w} \cup K_{t_{i+1},w}$ for $l \leqslant w < m$.

Case (iii) σ is not present in $\mathcal{F}_{t_{i+1}}$. In this case, we treat it as getting inserted at the maximum level L in $\mathcal{F}_{t_{i+1}}$ and hence, will be added by the inclusion $K_{t_{i+1},L} \hookrightarrow K_{t_i,L} \cup K_{t_{i+1},L}$.

We repeat this procedure for the pair $\mathcal{F}_{t_i}$ and $\mathcal{F}_{t_{i-1}}$.

Since we are working with an integer grid, we can simulate the quasi zigzag bi-filtration such that we have the standard filtration $\mathcal{F}_{t_i}$ at even x-coordinates and at odd x-coordinates, we have the standard filtration of the unions, without explicitly storing the unions.

Correctness. To prove the correctness of the procedure, we need to show the following: For $1 \leq i, i' \leq T$ and $1 \leq j, j' \leq L$, let $K_{i,j}$ denote the simplicial complex in the quasi zigzag filtration built according to the procedure mentioned above and $K'_{i,j}$ is the simplicial complex in the quasi zigzag bi-filtration built by considering unions explicitly. We show that the new simplices inserted by the inclusion $K_{i,j} \hookrightarrow K_{i',j'}$ are the same as the ones inserted by $K'_{i,j} \hookrightarrow K'_{i',j'}$.

Proof. The statement is immediately true for the inclusions $i' = i, j' = j + 1$. This is because the new simplices in the upward (y-direction) inclusions are the ones born at the level $j + 1$ in the input family of filtrations. Now, we prove the statement for the case where $i' = i \pm 1, j' = j$, i.e., the horizontal (x-direction) inclusions/deletions. Note that we have inclusions from $K_{i,j} \hookrightarrow K_{i',j'}$ only when i is an even integer and $i' = i \pm 1$. We prove it for the case $i' = i + 1, j' = j$. The proof for the other case is similar. Let σ be a simplex added on the inclusion $K_{i,j} \hookrightarrow K_{i+1,j}$. According to the procedure described above, it means that σ is born at some level $j_1 \leq j$ in the filtration $\mathcal{F}_{i+2}$ and is not born before level j in the filtration $\mathcal{F}_i$. Hence, σ will be newly added on the arrow $K'_{i,j} \hookrightarrow K'_{i+1,j}$ because $\sigma \in K'_{i+2,j}$ and $\sigma \notin K'_{i,j}$. This is because $K_{i,j} = K'_{i,j}$ for all even integers i, as they are the simplicial complexes part of the input filtrations $\mathcal{F}_i$. By the same arguments, all the newly added simplices on inclusions $K'_{i,j} \hookrightarrow K'_{i+1,j}$ will be newly added simplices on $K_{i,j} \hookrightarrow K_{i+1,j}$.

□

B.2 Algorithm to compute ZZ-GRIL

Algorithm 1 COMPUTEZZ-GRIL

Input: $\mathcal{ZZ}$: Quasi zigzag bi-filtration, $k \geq 1, \mathbf{p}$
Output: $\lambda(\mathbf{p}, k)$: ZZ-GRIL value at point $\mathbf{p}$ for fixed k
Initialize: $\delta_{min} \leftarrow 1, \delta_{max} \leftarrow \text{len}(\mathcal{F}), \lambda \leftarrow 1$
while $\delta_{min} \leq \delta_{max}$ **do**
  $\delta \leftarrow (\delta_{min} + \delta_{max})/2$
  $I \leftarrow \lfloor \mathbf{p} \rfloor_\delta^2$; $r \leftarrow$ COMPUTERANK($\mathcal{ZZ}, I$)
  if $r \geq k$ **then**
    $\lambda \leftarrow \delta$; $\delta_{min} \leftarrow \delta + 1$
  else
    $\delta_{max} \leftarrow \delta - 1$
  end if
end while
return λ

B.3 Time Complexity Analysis

Let N denote the length of the zigzag filtration at the top-most scale level, i.e., ($\mathcal{Z}_L$). For the analysis, we consider the zigzag filtration to be starting from an empty complex and ending at an empty complex. Assuming we have temporal data for T time steps and that $L < T$ where L denotes the number of different spatial scales, each iteration in the binary search requires $O(N^\omega)$ time since each iteration of the binary search involves computing zigzag persistence of a zigzag filtration of length $O(N)$. Here $\omega < 2.37286$ is the matrix multiplication exponent. Hence, the binary search in total requires $O(N^\omega \log T)$ time per center point. If we sample s center points, then the total time complexity would be $O(sN^\omega \log T)$.

C Experimental Details

Here, we report additional details about our experiments. We begin by including a detailed description of the UEA datasets in Table 5. Further, we have a comparison of treating a multivariate time series as a sequence of point clouds versus treating it as a sequence of graphs as the input to the ZZ-GRIL framework. We report the results of this experiment in Table 6. We can see that there is no clear winner. On some datasets, ZZ-GRIL extracts more relevant information from sequences of point clouds while on some datasets, information from sequences of graphs performs better. We have a visualization of ZZ-GRIL in Figure 7. We can see in the figure that the ZZ-GRIL topological signature is different for samples of different classes and very similar for samples in the same class.

For all the experiments with sequence of point clouds, we choose a window size that is $\max(5, series_length//128)$ and overlap $\max(4, 0.7 * window_size)$. For the experiments with sequence of graphs, we choose a window size of $\min(series_length/5, 128)$ and an overlap of $0.7 * window_size$. Then, from the complete graph, we randomly choose a percentage between 65 and 75 of the edges depending on their weights. This is to ensure that the number of edges does not remain the same for all the graphs in the sequence. For all our experiments, we use 36 center points to compute ZZ-GRIL at. For model parameters, mostly, we use the same parameters as specified by the respective models Liu et al. (2024); Zhao et al. (2023). We notice that the training is sensitive to learning rate and we optimize the learning rate between $1e-3$ and $5e-5$ for different datasets. These choice of hyperparameters are based on preliminary experiments, the results of which we report in Table 8, Table 9 , Table 10 and Table 11. The results reported here are for TodyNet+ZZ-GRIL model.

Dataset	Type	No. of series	Series length	No. of classes	Train size	Test size
FingerMovements	EEG	28	50	2	316	100
Heartbeat	AUDIO	61	405	2	204	205
MotorImagery	EEG	64	3000	2	278	100
NATOPS	HAR	24	51	6	180	180
SelfRegulationSCP2	EEG	7	1152	2	200	180

Table 5: Information about UEA Datasets used for experiments

Dataset	TodyNet + ZZ-GRIL point clouds	TodyNet + ZZ-GRIL graphs
FingerMovements	0.660	0.680
NATOPS	0.961	0.945
SelfRegulationSCP2	0.600	0.594

Table 6: Comparison of converting multivariate time series as a sequence of point clouds versus converting it as a sequence of graphs to extract the topological information.

Dataset	Snapshot PH	ZZ-GRIL
FingerMovements	0.480	0.590
Heartbeat	0.639	0.721
MotorImagery	0.560	0.580
NATOPS	0.777	0.850
SelfRegulationSCP2	0.466	0.522

Table 7: Comparison of vectorizing each non-zigzag filtration individually (Snapshot PH) versus ZZ-GRIL.

Dataset	36 center points	25 center points	64 center points
FingerMovements	0.660	0.620	0.680
SelfRegulationSCP2	0.600	0.578	0.600

Table 8: Selection of number of center points to compute ZZ-GRIL

Dataset	$\max(0.7 * \text{window_size}, 5)$	$\max(0.3 * \text{window_size}, 2)$	$\max(0.5 * \text{window_size}, 3)$
FingerMovements	0.660	0.620	0.640
SelfRegulationSCP2	0.600	0.550	0.578

Table 9: Selection of overlap between time windows for time-series modelled as a sequence of graphs. The $window_size$ is fixed to be $max(128//series_len, 5)$.

Dataset	$\max(\text{series_len}//128, 5)$	$\max(\text{series_len}//64, 10)$	$\max(\text{series_len}//256, 3)$
FingerMovements	0.660	0.640	0.580
SelfRegulationSCP2	0.600	0.578	0.550

Table 10: Selection of window size for time-series modelled as a sequence of graphs. The overlap is fixed to be $max(4, 0.7 * window_size)$.

Dataset	Threshold (45% - 55%)	Threshold (65% - 75%)	Threshold (85% - 95 %)
FingerMovements	0.600	0.660	0.630
SelfRegulationSCP2	0.550	0.600	0.550

Table 11: Selection of thresholding value range for retaining edges in the graph.

Dataset	Time (ZZ-GRIL point clouds)	Time (ZZ-GRIL graphs)
FingerMovements	569s	138s
NATOPS	413s	68s
SelfRegulationSCP2	153s	27s

Table 12: Computation times for ZZ-GRIL. The values represent the computation times for both training and testing splits. All the experiments were performed on Intel(R) Xeon(R) Gold 6248R CPU and NVIDIA Quattro RTX 6000 GPU.

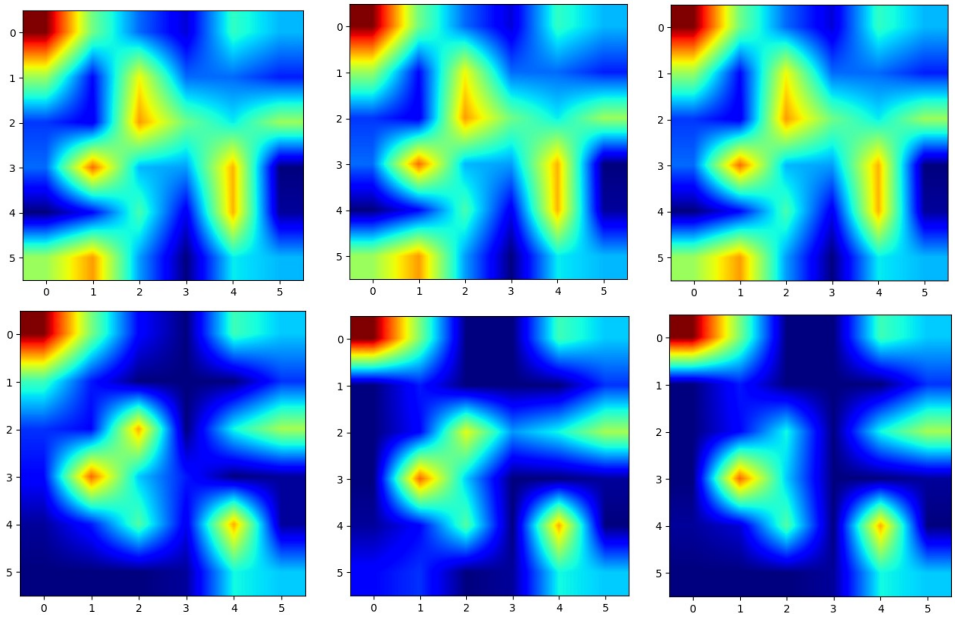

Figure 7: Heatmap of ZZ-GRIL on the FingerMovements dataset. ZZ-GRIL is computed at 36 center points, which is plotted as a 6x6 grid. The top row represents three samples with label 1 and the bottom row represents three samples with label 0. We can see the similarity in the ZZ-GRIL signatures between samples of the same class and clear differences between samples of different classees.