The sublevel Flood bifiltration: towards scalable 2-parameter persistent homology

Mattéo Clémot
Université Lyon 1
matteo.clemot@univ-lyon1.fr

Julie Digne
CNRS, Université Lyon 1
julie.digne@cnrs.fr

Julien Tierny
CNRS, Sorbonne Université
julien.tierny@cnrs.fr

Abstract

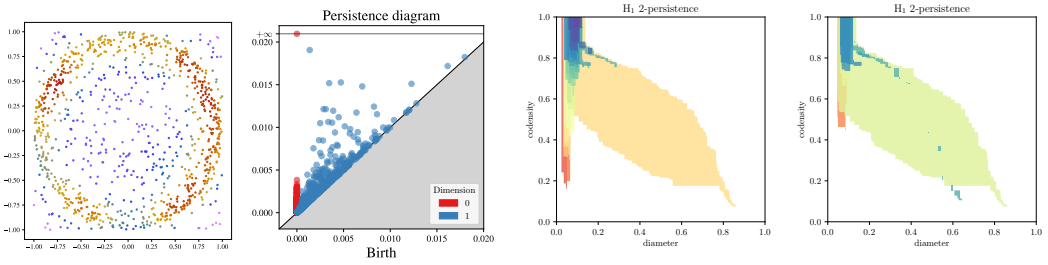
Multiparameter persistent homology is a rapidly developing branch of topological data analysis that improves the robustness of single-parameter persistent homology to outliers, while still capturing the metric characteristics of the data. However, a notable limitation is its lack of scalability. In this paper, we introduce a novel approach for efficiently computing 2-parameter persistent homology on large point sets. Our work extends the Flood filtration, originally developed for single-parameter persistence. Our construction, called the *sublevel Flood bifiltration*, offers a scalable approximation of the sublevel offset bifiltration. We show that it benefits from theoretical stability properties and describe how to compute it efficiently. We demonstrate the performance of our approach in classification tasks on low-dimensional synthetic datasets, where density awareness is critical, as well as on real-world time series datasets.

1 Introduction

Topological data analysis (TDA) has lately emerged as a toolbox to handle geometric and topological properties of data, using ideas from algebraic topology. It provides concise and stable representations encoding topological information that is often not captured by classical approaches. It has been successfully used in machine learning, in a variety of domains and tasks, including dimensionality reduction [52, 61], classification [57] or AI interpretability [32, 31] (see [36] for a recent survey). One of the main tools of TDA is *persistent homology* (PH), which considers *filtrations*, typically nested sequences of simplicial complexes that represent the data at different scales. The evolution of their topological features (connected components, loops, cavities) as the scale parameter increases is then summarized into concise and stable representations, like the *persistence barcode* or the *persistence diagram*. This single-parameter framework is now mathematically well-understood [29].

Single-parameter persistent homology For the analysis of Euclidean point sets, common filtrations include the Rips and Čech filtrations. However, as their number of simplices increases rapidly with the number of vertices (exponentially in general, cubically when considering only their 2-skeletons for 1-dimensional PH), these filtrations become computationally intensive both in memory and runtime, which often limits the size of the input. To address this limitation, several approaches have been proposed: algorithmic optimization [5] with efficient, parallel implementation on CPU [56] or GPU [64]; sparse linear-size approximations [59]; edge collapses for reducing flag filtrations [12, 33]; and for low-dimensional Euclidean point sets, Delaunay-based filtrations (such as Delaunay, Delaunay–Čech [7] and Delaunay–Rips [51, 23] filtrations) which feature far fewer simplices. More recently, Graf et al. introduced the *Flood filtration* [35]. For an input point set X, this filtration is built on the Delaunay complex of a smaller set of landmarks L, yet it accounts for the entire set X, since each simplex is introduced as soon as its convex hull is fully covered by balls of radius r centered at points in X. It is stable under perturbations of X, and, in $\mathbb{R}^2$, is proven to approximate the offset filtration of X. In practice, this approach allows the computation of filtration values to be

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) (b) (c) (d)

Figure 1: **(a)** A point set $X \subset \mathbb{R}^2$ colored by its kernel density estimate (KDE). **(b)** The persistence diagram of its Delaunay–(Čech) filtration captures many small cycles (blue dots) with similar persistence (between 0.01 and 0.02). **(c)** Multiparameter module approximation (MMA) of the sublevel Delaunay bifiltration. **(d)** MMA of our sublevel Flood bifiltration (100 landmarks). The yellow (c) and green (d) large shapes both distinctively capture the prominent ring (a) with high density and large diameter (up to 0.8).

parallelized on GPUs, resulting in significant acceleration compared to the Delaunay–Čech PH, for large point sets.

Multiparameter persistent homology Single-parameter PH, especially for diameter-based filtrations, is highly sensitive to outliers. To mitigate this limitation, while still capturing the metric properties of the input data, one can consider more than one parameter, for instance by filtering with respect to both the geometric scale and some density estimate (see Fig. 1). This has given rise to multiparameter PH [15], which turns out to be more challenging mathematically and computationally.

Bifiltrations One way to define a density-aware bifiltration is to introduce a second (discrete) parameter that encodes some notion of degree or connectivity. Examples include the degree-Rips and degree-Čech bifiltrations [45], the multicover bifiltration [30, 27] and its sparse linear-size approximations [2], and the Core and Delaunay–Core bifiltrations [10]. These constructions are all interleaved and benefit from stability properties with respect to the Prohorov distance [11]. Another approach is to build a bifiltration over a point set X using an explicit function $\gamma : X \to \mathbb{R}$ accounting for the point set density. γ is typically an estimate of codensity (i.e., the opposite of a density estimation, for instance obtained by kernel density estimation (KDE) or by distance-to-a-measure (DTM) [19, 21]) but can also be an estimation of eccentricity, of curvature, or even an application-provided function (e.g., the partial charge of atoms in a molecule [39]). This offers more flexibility, but at the cost of no longer being parameter-free. In this setting, the theoretical reference bifiltration is the *sublevel offset bifiltration*, which, at scale (r, s), consists of the union of balls of radius r centered at points $x \in X$ such that $\gamma(x) \leq s$, i.e., $\mathcal{O}_{r,s}(X, \gamma) = \bigcup_{x \in X_s} B_r(x)$ where $X_s = \gamma^{-1}((-\infty, s])$ (see Fig. 2). In practice, one can consider the *sublevel Rips bifiltration* [13] given by $\mathcal{R}_{r,s}(X, \gamma) = \mathcal{R}_r(X_s)$. In low-dimensional Euclidean spaces, the *sublevel Delaunay bifiltration* [3] is far smaller and far more efficient. It constructs a bifiltration on the *incremental Delaunay complex* $\mathcal{I}(\gamma)$, which solves the non-monotonicity issue of the Delaunay complex, and is topologically equivalent to the sublevel offset bifiltration.

Representations Applying the homology functor on a bifiltration yields its associated multiparameter persistence module. Unlike the single-parameter case, the multiparameter framework lacks a complete, discrete invariant analogous to the persistence barcode [15]. This is because multiparameter modules cannot generally be decomposed into a direct sum of interval modules, except in some very specific cases [14, 25]. For instance, the *fibered barcode* [45] of a multiparameter persistence module M, which consists of the collection of (single-parameter) persistence barcodes of 1-dimensional affine slices of M, does not fully describe M. Several *candidate decomposition* methods [4, 48] have been proposed to alleviate this by approximating M as a direct sum of interval modules. To incorporate multiparameter PH indicators into machine learning pipelines, various vectorizations have been proposed. These include the multiparameter persistence kernel (MPK) [26] and landscapes (MPL) [62] that are both defined on the fibered barcode. Multiparameter persistence images (MPI) [16] leverage

vineyards to enhance discrimination between modules that share the same fibered barcode. The stable candidate decomposition representation (SCDR) [47] relies on candidate decomposition and comes with stability guarantees. Finally, graphcodes [43] enable vectorization-free learning from multiparameter persistent homology.

Contributions Current approaches for computing 2-parameter persistent homology prove too computationally intensive for practical use. In this work, we tackle this challenge by introducing the *sublevel Flood bifiltration*, along with its theoretical foundations and practical algorithm. This construction is significantly smaller than existing bifiltrations and leads to orders-of-magnitude speedups for downstream operations (e.g., module candidate decomposition and vectorization). More precisely, our contributions are as follows:

- we introduce the *sublevel Flood bifiltration*, establish its stability properties, and empirically assess its approximation of the sublevel offset bifiltration;
- we propose an efficient, GPU-friendly computational approach for this bifiltration (our source code is available at `https://github.com/MClemot/SLFlood` and a Python package is available on PyPI at `https://pypi.org/project/sublevel-flood/`);
- we demonstrate, through classification experiments on both synthetic and real-world time series datasets, that our method matches the performance of existing sublevel-based bifiltrations while being far more compact.

2 Background

We consider a point set X in $\mathbb{R}^d$, with d small enough (typically, $d \leq 10$) to allow for the computation of the Delaunay complex.

We quickly recall below the foundations of (single-parameter) persistent homology and refer the reader to [29, 22] for further details. A *filtration* is a nested sequence of simplicial complexes $\mathcal{K} = (\mathcal{K}_r)_{r\geq 0}$, i.e., such that for any $r \leq r'$, $\mathcal{K}_r \subseteq \mathcal{K}_{r'}$. The k-th homology group $H_k(\mathcal{K}_r)$ is a vector space representing the k-dimensional topological features of $\mathcal{K}_r$ (i.e., connected components for $k = 0$, loops for $k = 1$, cavities for $k = 2$). The core idea of persistent homology is to apply the homology functor to this sequence of simplicial complexes: as r increases, the topological features emerge and vanish. This information is summarized as the multiset $\mathrm{dgm}_k(\mathcal{K}) = \{(b_i, d_i) \mid i \in [\![1, m]\!]\}$ where b_i and d_i are the filtration values at which a k-dimensional topological feature is born and dies, respectively. Algebraically, this corresponds to the persistence module $\mathbb{K}_k = (H_k(\mathcal{K}_r))_{r\geq 0}$ (with linear maps induced by simplicial inclusions) admitting a decomposition into a direct sum of interval modules: $\mathbb{K}_k = \bigoplus_{i=1}^{m} \mathbb{I}[b_i, d_i]$ [20].

A notable filtration is the *offset filtration* $\mathcal{O}_r(X) = \bigcup_{x\in X} B_r(x)$. Due to its non-combinatorial nature, one usually rather considers its nerve, known as the Čech filtration $\check{\mathcal{C}}_r(X) = \left\{\sigma \subset X \mid \bigcap_{x\in\sigma} B_r(x) \neq \varnothing\right\} = \mathrm{nrv}\left((B_r(x))_{x\in X}\right)$. In low-dimensional Euclidean spaces, it is computationally more efficient to use the alpha (or Delaunay) filtration defined as $\mathcal{D}_r(X) = \mathrm{nrv}\left((B_r(x) \cap \mathrm{Vor}(x, X))_{x\in X}\right)$ which is homotopy equivalent to $\mathcal{O}_r(X)$.

2.1 The Flood filtration

The Flood complex was introduced in [35] to improve the scalability of PH when the size of X increases. It relies on a set of landmarks L that is often—but not necessarily—a subset of X, e.g., via furthest point sampling. At scale r, it contains all simplices in the Delaunay complex $\mathrm{Del}(L)$ that are covered by the union of balls of radius r, centered at the points of X:

$$\mathcal{F}_r(X, L) = \left\{\sigma \in \mathrm{Del}(L) \mid \mathrm{conv}(\sigma) \subset \bigcup_{x\in X} B_r(x)\right\} \tag{1}$$

where $\mathrm{conv}(\sigma)$ is the convex hull of σ. The persistence diagrams of the Flood filtration are bottleneck-stable with respect to X, i.e., for any X, Y, L and any $k \in \mathbb{N}$:

$$\mathrm{d_b}\left(\mathrm{dgm}_k(\mathcal{F}(X, L)), \mathrm{dgm}_k(\mathcal{F}(Y, L))\right) \leq \mathrm{d_H}(X, Y) \tag{2}$$

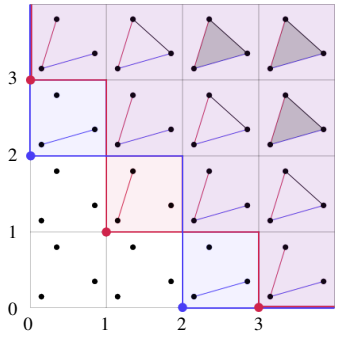
where $\mathrm{d_b}$ is the bottleneck distance between persistence diagrams, and $\mathrm{d_H}$ is the Hausdorff distance. Besides, in the specific case where $L, X \subset \mathbb{R}^2$, it approximates the topology of the offset filtration:

$$\forall k \in \mathbb{N}, \mathrm{d_b}\left(\mathrm{dgm}_k(\mathcal{F}(X,L)), \mathrm{dgm}_k(\mathcal{D}(X))\right) \leq 2\,\mathrm{d_H}(X,L). \tag{3}$$

This guarantees that the approximation of the topology of $\mathcal{D}(X)$ by the Flood filtration is at most twice the error induced by a subsampling strategy (i.e., computing the PH of $\mathcal{D}(L)$). In practice, however, the approximation turns out to be significantly better than this theoretical bound. This improvement arises because the filtration accounts for the entire X (through the consideration of the union of balls centered at X), despite being defined only on $\mathrm{Del}(L)$.

2.2 2-parameter persistent homology

Bifiltrations A *bifiltration* is a simplicial complex $\mathcal{K}$ and a collection of subcomplexes $(\mathcal{K}_\alpha)_{\alpha\in\mathbb{R}^2}$ that is increasing in the sense that $\mathcal{K}_\alpha \subseteq \mathcal{K}_\beta$ whenever $\alpha \preceq \beta$ (which means $\alpha_1 \leq \beta_1$ and $\alpha_2 \leq \beta_2$). The *support* of a simplex $\sigma \in \mathcal{K}$ is $\mathrm{supp}(\sigma) = \{\alpha \in \mathbb{R}^2 \mid \sigma \in \mathcal{K}_\alpha\}$. This general definition can be adapted to our specific context, where, informally, each simplex is *inserted* into $\mathcal{K}$ along a *staircase* (i.e., each support is a finite union of upper-right quadrants). More formally, each simplex has a finite, minimal set of *bigrades* $\phi(\sigma) \subset \mathbb{R}^2$ such that $\sigma \in \mathcal{K}_\alpha$ if and only if there exists $\beta \in \phi(\sigma)$ such that $\beta \preceq \alpha$. This minimal set of bigrades $\phi(\sigma)$ corresponds to the Pareto front[1] of $\mathrm{supp}(\sigma)$ and represents the multiple appearances of σ within $\mathcal{K}$ along the staircase. This is analogous to the simplex *insertion* in single-parameter persistence. The bifiltration is said to be *k-critical* if every $\sigma \in \mathcal{K}$ has a minimal set of bigrades of size $|\phi(\sigma)| \leq k$. For instance, the sublevel Rips and Delaunay bifiltrations are always 1-critical while the degree-Rips bifiltration is multi-critical. The inlined figure illustrates a 3-critical example, where the support of the red (respectively blue) edge of $\mathcal{K}$ is shaded in red (resp. blue) with 3 (resp. 2) bigrades marked as red (resp. blue) dots at the corners of the support.



Multiparameter persistence modules Applying the k-th homology functor H_k on a bifiltration yields the associated 2-parameter persistence module $\mathbb{K} = (H_k(\mathcal{K}_\alpha))_{\alpha\in\mathbb{R}^2}$. It can be viewed as a collection of vector spaces equipped with linear maps $H_k(\mathcal{K}_\alpha) \to H_k(\mathcal{K}_\beta)$ for each $\alpha \preceq \beta$ induced by the simplicial inclusions $\mathcal{K}_\alpha \subset \mathcal{K}_\beta$. The interleaving distance $\mathrm{d_I}$ between (isomorphism classes of) 2-parameter persistence modules generalizes from the single-parameter case [44]. However, its computation is NP-hard [8]. A more practical alternative is the matching distance $\mathrm{d_{match}}$ [17], a distance between fibered barcodes, providing a lower bound for the interleaving distance (see Appendix A for details).

Module preprocessing A common preprocessing step is to compute a *minimal presentation* [46, 42], which reduces the size of the input by providing the smallest possible representation of a multiparameter persistence module. This often accelerates subsequent algorithms, such as candidate decomposition. However, these minimal presentation algorithms require *free*, i.e., *1-critical*, bifiltrations. To handle multi-critical bifiltrations, [18] proposed an algorithm to compute a *free implicit representation*. More recently, [6] proposed two *one-criticalization* algorithms that both compute a *free resolution* of the input, i.e., a free chain complex that is isomorphic in homology to the input.

See Appendix A for further background on multiparameter persistent homology.

3 The sublevel Flood bifiltration

This section presents our novel bifiltration for the efficient computation of 2-parameter persistent homology. Proofs are provided in Appendix B.

[1] The Pareto front of $S \subset \mathbb{R}^2$ is defined as $\mathrm{Pareto}(S) = \{\alpha \in S \mid \nexists \beta \in S, \beta \preceq \alpha \text{ and } \alpha \neq \beta\}$.

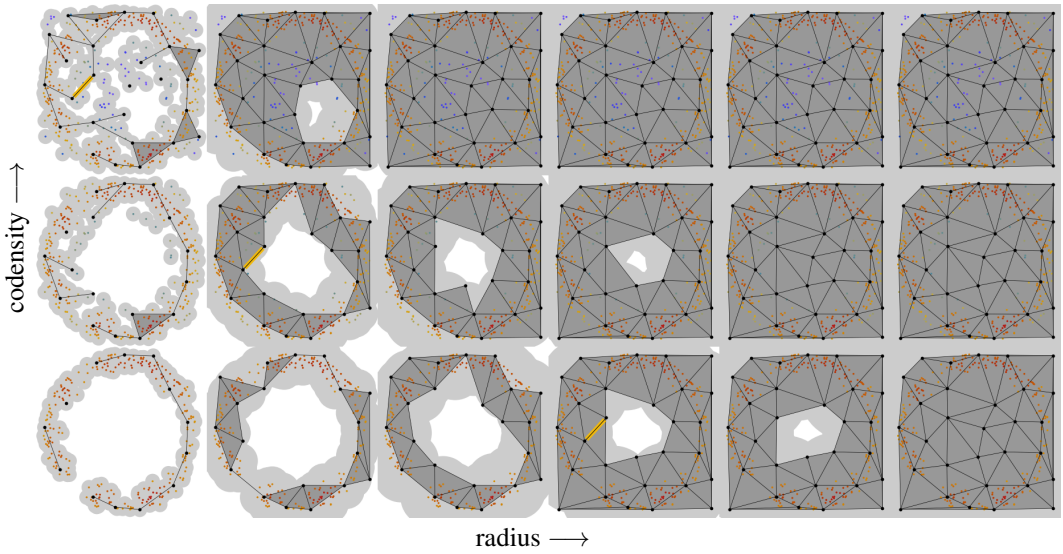


Figure 2: $\mathcal{F}_{r,s}(X, L, \gamma)$ is represented with black vertices, black edges and dark gray triangles for $L \subset X \subset \mathbb{R}^2$ ($|X| = 400$, $|L| = 30$), and $\gamma : X \to \mathbb{R}$ a codensity function obtained by a Gaussian kernel density estimator. The edge highlighted in yellow illustrates the multi-criticality of the construction, as its minimal set of bigrades is of size at least 3. The sublevel offset bifiltration $\mathcal{O}_{r,s}(X, \gamma) = \mathcal{O}_r(X_s)$ is shown in light gray.

3.1 Definition

In order to extend the Flood filtration to the 2-parameter framework, we define the *sublevel Flood bifiltration* for $X, L \subset \mathbb{R}^d$ and a function $\gamma : X \to \mathbb{R}$, writing $X_s = \gamma^{-1}((-\infty, s])$, as follows:

$$\mathcal{F}_{r,s}(X, L, \gamma) = \mathcal{F}_r(X_s, L) = \left\{ \sigma \in \mathrm{Del}(L) \mid \mathrm{conv}(\sigma) \subset \bigcup_{x \in X_s} B_r(x) \right\}. \tag{4}$$

This bifiltration is multi-critical but uses a small complex, in contrast to sublevel Rips and Delaunay bifiltrations which are 1-critical but require a large complex to describe the data. Fig. 2 illustrates this construction on a planar point set.

We suppose in the following that $X = (x_1, \ldots, x_n)$ is sorted in increasing order of γ. We also write for convenience $\gamma_i = \gamma(x_i)$ and $X_{\leq i} = X_{\gamma_i} = (x_1, \ldots, x_i)$. The support of a simplex $\sigma \in \mathrm{Del}(L)$ is then described by the following set of bigrades:

$$\phi(\sigma) = \{(\mathrm{fl}(\sigma, X_{\leq i}), \gamma_i), 1 \leq i \leq n\} \text{ with } \mathrm{fl}(\sigma, X) = \max_{p \in \mathrm{conv}\,\sigma} \mathrm{d}(p, X) = \max_{p \in \mathrm{conv}\,\sigma} \min_{x \in X} \mathrm{d}(p, x) \tag{5}$$

where d is the Euclidean distance. Taking the Pareto front of this set gives the minimal set of bigrades of σ. In the worst case, its size can reach $|X|$, which means that $\mathcal{F}(X, L, \gamma)$ is, in general, $|X|$-critical. In practical cases, however, the criticality turns out to be much smaller.

In the following, we write $\mathbb{F}(X, L, \gamma)$ for the 2-parameter persistence module obtained by applying the homology functor to the Flood bifiltration: $\mathbb{F}(X, L, \gamma) = (H_*(\mathcal{F}_{r,s}(X, L, \gamma)))_{r,s}$ with the maps induced by the simplicial inclusions.

3.2 Properties

A desirable property for a topological indicator is stability under input perturbation. We quantify this stability by providing an upper bound on the interleaving distance between the sublevel Flood persistence module of X and that of its perturbation Y, with a fixed set of landmarks. This, in turn, implies the same bounds for the matching distance between the associated fibered barcodes.

Theorem 1 (Stability w.r.t. X and γ)**.** *Let $X, Y \subset \mathbb{R}^d$ be two point sets. Suppose that both γ_X and $\gamma_Y : \mathbb{R}^d \to \mathbb{R}$ are ℓ-Lipschitz. With L fixed, the interleaving distance between both sublevel Flood*

modules is upper bounded by:

$$\mathrm{d_I}\left(\mathbb{F}(X,L,\gamma_X),\mathbb{F}(Y,L,\gamma_Y)\right) \leq \max\left(\mathrm{d_H}(X,Y), \ell\,\mathrm{d_H}(X,Y) + \|\gamma_X - \gamma_Y\|_\infty\right). \tag{6}$$

As mentioned earlier, γ is often a codensity estimation obtained by KDE or DTM. In this case, we can refine the stability result when each point of X is displaced.

Theorem 2 (Stability w.r.t. X)**.** *Let $X \subset \mathbb{R}^d$ be a point set and Y its ε-perturbation such that $\|x_i - y_i\|_2 \leq \varepsilon$ for all i. We consider the case where L is fixed. If γ_X and γ_Y are obtained by a KDE with an ℓ_K-Lipschitz kernel K and bandwidth h, then:*

$$\mathrm{d_I}\left(\mathbb{F}(X,L,\gamma_X),\mathbb{F}(Y,L,\gamma_Y)\right) \leq \varepsilon \max\left(1, \frac{2\ell_K}{h^{d+1}}\right).$$

If γ_X and γ_Y are obtained by DTM with parameter $m \in (0,1]$, then:

$$\mathrm{d_I}\left(\mathbb{F}(X,L,\gamma_X),\mathbb{F}(Y,L,\gamma_Y)\right) \leq \varepsilon\left(1 + \frac{1}{\sqrt{m}}\right).$$

Thus, if the input X is slightly perturbed while L remains fixed, the associated persistence module changes only slightly (with respect to the interleaving distance), as do its vectorizations, whether they are based on the fibered barcode or finer representations like SCDR.

3.3 Computational approach

Landmark selection As in [35], we want to minimize $\mathrm{d_H}(X,L)$ to ensure a well-distributed set of landmarks. However, given a fixed number of landmarks $|L|$, finding the optimal $L \subset X$ that minimizes $\mathrm{d_H}(X,L)$ amounts to solving a metric $|L|$-center problem which is NP-hard [37]. Instead, we adopt below the same greedy strategy as in [35], based on furthest point sampling. See Appendices D.5 and D.6 for discussions on the sampling strategy and the number of landmarks.

Masking strategy A key aspect of the computational efficiency of the Flood filtration is its local nature: the filtration value $\mathrm{fl}(\sigma)$ of a simplex $\sigma \in \mathrm{Del}(L)$ is not affected by points that are sufficiently far from σ. More precisely, if $B_r(c)$ is an enclosing ball of σ, then $\mathrm{fl}(\sigma) = \max_{p\in\mathrm{conv}\,\sigma} \mathrm{d}(p, X \cap B_{\sqrt{2}r}(c))$. This allows for a masking strategy in which each simplex only considers the points of X that are in the $\sqrt{2}$-dilation of its minimal enclosing ball (meb). Unfortunately, this approach does not directly extend to the sublevel Flood bifiltration, since a point with a low γ value may influence the set of bigrades of arbitrarily far simplices. To keep the number of points per simplex manageable, we resort to the following approximation. We first compute the minimal set of bigrades $\phi(\{l\})$ of each vertex $l \in L$, and write $X^{\{l\}} \subseteq X$ for the points that contribute to it.[2] Now, let $\sigma \in \mathrm{Del}(L)$. We define the set of points that are considered for determining its bigrades as

$$X^\sigma = \bigcup_{\tau\in\mathrm{C}(\sigma)} \left(X \cap B_{\sqrt{2}r_\tau}(c_\tau)\right) \cup \bigcup_{l\in\mathrm{V}(\sigma)} X^{\{l\}},$$

where $\mathrm{C}(\sigma)$ contains the maximal cofaces of σ, $B_{r_\tau}(c_\tau)$ is the minimal enclosing ball of τ, and $\mathrm{V}(\sigma)$ contains the vertices appearing in the cofaces of σ.[3] Informally, the left union includes points that are *near* σ, which impact the set of bigrades $\phi(\sigma)$ for low diameters and large γ values. The right union, on the other hand, contains *farther points* that typically impact $\phi(\sigma)$ for large diameters and low γ values. We write:

$$\widetilde{\mathcal{F}}_{r,s}(X,L,\gamma) = \{\sigma \in \mathrm{Del}(L) \mid \mathrm{conv}(\sigma) \subset \bigcup_{x\in X_s^\sigma} B_r(x)\} \text{ with } X_s^\sigma = \{x \in X^\sigma \mid \gamma(x) \leq s\}. \tag{7}$$

Theorem 3. *$\widetilde{\mathcal{F}}(X,L,\gamma)$ is a bifiltration and, in addition, noting $\Delta(L) = \max_{\sigma\in\mathrm{Del}(L)} \delta(\sigma)$ where δ is the diameter, we have $\mathrm{d_I}(\mathbb{F}(X,L,\gamma), \widetilde{\mathbb{F}}(X,L,\gamma)) \leq 2\Delta(L)$.*

[2] i.e., $X^{\{l\}} = \{x \in X \mid \nexists y \in X \text{ s.t. } y \neq x \text{ and } \gamma(y) \leq \gamma(x) \text{ and } \mathrm{d}(y,l) \leq \mathrm{d}(x,l)\}$.

[3] i.e., $\mathrm{V}(\sigma) = \{l \in L \mid \exists \tau \text{ s.t. } \sigma \subset \tau \text{ and } l \in \tau\}$.

While this result suggests that a dense set of landmarks is needed to benefit from the properties of the Flood bifiltration with our approximate masking strategy, this bound is overly pessimistic, because it does not account for the inclusion of all *near* points in X^σ. See Fig. 4 for a practical evaluation of the ability of this masked Flood bifiltration to approximate the sublevel offset bifiltration, and Fig. 10 for an evaluation of its stability with respect to perturbations of the input. See also Appendix D.8 and Table 5 for a comparison between the exact (unmasked, Eq. (4)) and approximate (masked, Eq. (7)) sublevel Flood bifiltrations in terms of approximation and runtime.

Simplex discretization and computation As in the original Flood algorithm, we discretize each convex hull $\mathrm{conv}(\sigma)$ by a discrete point set $P_\sigma \subset \mathrm{conv}(\sigma)$, which is chosen on a grid obtained with evenly spaced barycentric coordinates. Then, when computing the set of bigrades of $\sigma \in \mathrm{Del}(L)$ (see Eq. (5)), $\max_{p\in\mathrm{conv}\,\sigma} \mathrm{d}(p, X_s^\sigma)$ is replaced by $\max_{p\in P_\sigma} \mathrm{d}(p, X_s^\sigma)$. The error introduced by this approximation is controlled by the number of grid points on each simplex edge (see [35], Theorem 5 and Lemma 6, for a detailed discussion). Thus, with X^σ still ordered by increasing value of γ, the set of bigrades $\phi(\sigma)$ is computed as the Pareto front of:

$$\left\{ \left(\max_{p\in P_\sigma} \min_{x\in X^\sigma_{\leq i}} \mathrm{d}(p,x), \gamma(X_i^\sigma) \right), 1 \leq i \leq |X^\sigma| \right\}.$$

Concretely, given the distance matrix $D_\sigma = \mathrm{d}(P_\sigma, X^\sigma)$, one can compute $\phi(\sigma)$ by first applying a cumulative minimum operation along each row, and then taking the maximum on each column. We implemented this as batched operations on GPU tensors. See Fig. 7 in Appendix C for an illustration. The speedup obtained from GPU acceleration is discussed in Appendix D.7 and illustrated in Fig. 16.

4 Experiments

We implemented our algorithm in Python using PyTorch [53] to compute bifiltration values with tensors on the GPU. For comparisons, we used the multipers library [49] for the multiparameter module approximation [48], vectorizations [47, 62], and wrappers of multiparameter PH algorithms (sublevel Delaunay bifiltrations [3] and minimal presentations [42]). Diameter-based and density-based persistence diagrams were computed and vectorized with the Gudhi library [50]. Finally, the machine learning pipelines were set up using the scikit-learn library [54]. In all our experiments, we sample 20 points on each simplex edge to define the barycentric grids P_σ (see Section 3.3; an edge has 20 samples, a triangle 210 and a tetrahedron 1540). Timing results were obtained on a server with 12 CPU cores, 64 GB of RAM and an H100 GPU. See Appendix D for additional experiments.

4.1 Runtime

In Fig. 3, we compare the running times for computing the bifiltration plus its MMA, with an optional intermediate minimal presentation (MP) step for the sublevel Delaunay and Rips bifiltrations. For our sublevel Flood bifiltration, we test several numbers of landmarks $|L| \in \{100, 300, 500, 1000\}$. We run these experiments for $d = 2$ and $d = 3$: in both cases, half of the points in $X \subset \mathbb{R}^d$ are distributed over the unit $(d-1)$-sphere with added Gaussian noise, while the other half are uniformly distributed inside the d-ball of radius $3/2$. We show the running time of the bifiltration construction alone (without MMA) in Fig. 8. We also report in Fig. 9, the sizes of the constructions: total number of bigrades (which coincides with the number of simplices for 1-critical bifiltrations); number of generators after minimal presentation; number of summands after MMA. This shows that our construction offers a clear speedup for computing an MMA from a point set compared to the Delaunay bifiltration, when X is sufficiently large. In Fig. 3, using the Flood bifiltration becomes worthwhile starting from $|X|$ between 200 and 3,000 depending on the dimension d and the number of landmarks. As $|X|$ continues to grow, the computation time scales more favorably than for other methods: this is due to the fixed size of the complex on which the bifiltration is built.

4.2 Approximation and stability

Fig. 4 assesses the ability of the sublevel Flood bifiltration to estimate the topology of the reference sublevel offset bifiltration, by reporting the matching distance (Appendix A) $\mathrm{d}_{\mathrm{match}}\left(H_k(\mathcal{F}(X,L,\gamma)), H_k(\mathcal{D}(X,\gamma))\right)$ for $k \in \{0,1,2\}$ when the size of L increases until $L = X$.

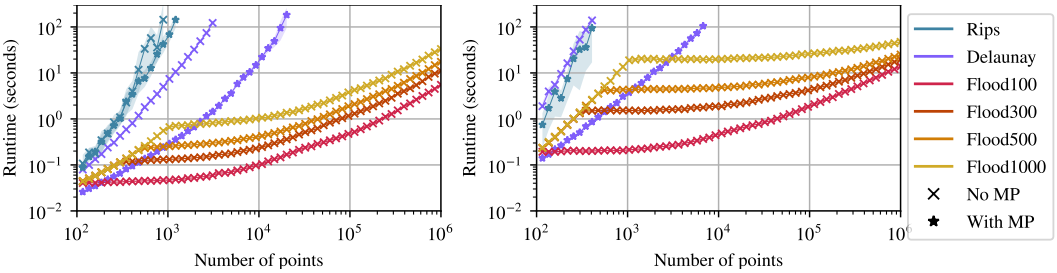


Figure 3: Running time of the bifiltration computation and its multiparameter module approximation. The homology is computed for degrees $k \in \{0, 1\}$ in $\mathbb{R}^2$ (left) and degrees $k \in \{0, 1, 2\}$ in $\mathbb{R}^3$ (right).

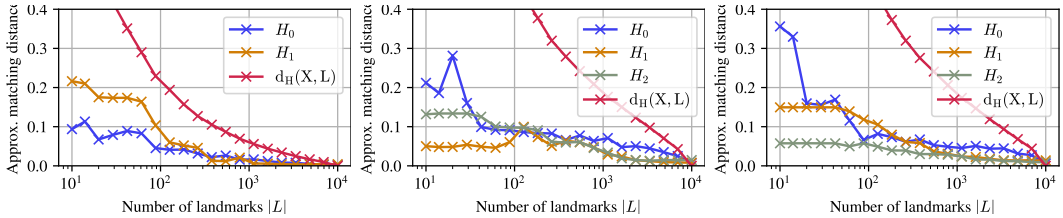


Figure 4: Matching distance between the modules of sublevel Flood and sublevel Delaunay bifiltrations, when the number of landmarks $|L|$ increases. X has 10,000 points distributed over the noisy ring in $\mathbb{R}^2$ (left), the noisy sphere in $\mathbb{R}^3$ (center) and the noisy torus in $\mathbb{R}^3$ (right).

We also evaluate in Fig. 10 the empirical stability of the construction with respect to the input X by computing $\mathrm{d}_{\mathrm{match}}\left(\mathbb{F}(X, L_X, \gamma_X), \mathbb{F}(Y, L_Y, \gamma_Y)\right)$ with $\|x_i - y_i\| \leq \varepsilon$, both when $L_X = L_Y$ is fixed and when L_X and L_Y are computed independently from X and Y.

4.3 Classification problems

To demonstrate the practical value of the sublevel Flood bifiltration, we compare its performance against other bifiltrations and single-parameter homology in classification tasks. The specifics of our datasets are provided in Appendix E.

For bifiltrations, we vectorize the multiparameter module approximation (MMA) into either stable candidate decomposition representations (SCDR) [47] or multiparameter persistence landscapes (MPL) [62]. To evaluate the interest of the multiparameter framework, we also compare to single-parameter filtrations: a diameter-based filtration (Delaunay–Čech) and a density-based filtration (sublevel filtration of γ evaluated on a regular cubical complex in the ambient space). The obtained persistence diagrams are vectorized using the ATOL method [58] (respectively "diameterPD" and "densityPD"). We also consider the concatenation of both vectorizations ("bothPD"). All vectorizations are given to a LightGBM [38] classifier with default parameters. Hyperparameters are validated with 5-fold cross-validation, using the Bayesian optimization framework Optuna [1] with 50 trials. For SCDR, we cross-validate the power $p \in \{0, 1\}$, the bandwidth $h \in [10^{-3}, 10^{0}]$ on a log-scale, and the resolution in $\{10 \times 10, 20 \times 20, 50 \times 50\}$. For MPL, only the resolution is cross-validated in the same range. For ATOL, the number of k-means centroids is selected in $\{32, 64, 128\}$ and the contrast function ([58], Eq. 1) is selected in $\{\text{Gaussian}, \text{Laplacian}, \text{indicator}\}$.

Toy synthetic dataset (Fig. 5, left, and Table 1) This dataset contains point sets with lower-density regions of varying sizes. The goal is to classify the point sets based on the diameter of these lower-density areas, where a difficulty parameter controls how well-separated the classes are. On this task, 2-parameter persistence methods outperform single-parameter ones, for all difficulty parameters. This suggests that they capture meaningful topological information that is not captured by single-parameter PH. Besides, except at the highest difficulty level, the gap in accuracy score between Delaunay and Flood bifiltrations is negligible, and the number of landmarks has little influence on the score: this confirms that, in this task, the classification accuracy mostly depends on the 2-parameter persistence features that exist for large diameter parameters r. These features do not require a particularly fine complex to be captured, as they correspond to the low-density regions.

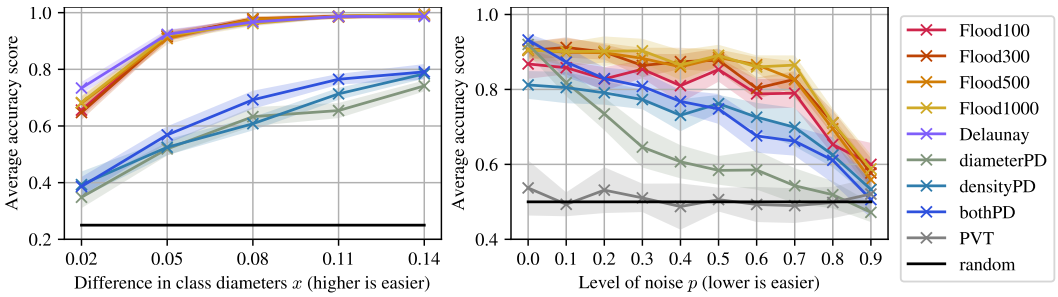


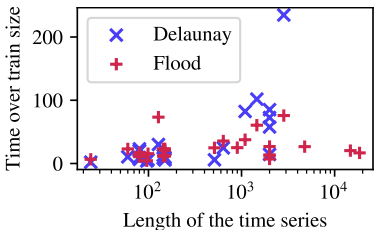
Figure 5: Accuracy scores (averaged over 10 splits) of single-parameter PH (diameterPD, densityPD and bothPD) and multiparameter PH (Flood and Delaunay). **Left:** Synthetic noisy swisscheese toy datasets in $\mathbb{R}^2$ parameterized by difficulty x. Here, the dataset contains 800 sets equally split into four classes characterized by the diameter of their low-density regions. Each set contains 5,000 points. **Right:** Synthetic porous material datasets on a 64^3 grid. The problem difficulty increases with the parameter p that quantifies the level of ambient noise. We also show the result obtained with the neural network-based method PVT [63] for reference.

Table 1: Timing in seconds for the noisy swisscheese datasets with 5,000 and 20,000 points (`NSC-*D-*k`) and the porous rocks dataset (`rocks64`), averaged over all their difficulties. The left number is the computation time of MMAs of the whole dataset. The right number is the classification time from the MMAs (averaged over the 10 splits). OOM means that memory was exceeded. A point set in `rocks64` has on average 69,000 points for $p = 0$ and 130,000 points for $p = 0.9$.

Dataset	Flood100	Flood300	Flood500	Flood1000	Delaunay
`NSC-2D-5k`	38.3 / 17.4	85.3 / 32.3	142.7 / 49.1	312.6 / 57.8	1484.6 / 91.8
`NSC-2D-20k`	91.3 / 11.5	150.2 / 12.9	219.3 / 19.4	416.4 / 38.3	OOM
`NSC-3D-5k`	183.9 / 24.8	800.7 / 43.6	2309.3 / 53.0	13990.6 / 130.3	OOM
`rocks64`	602.7 / 11.3	1086.7 / 16.1	1815.3 / 41.8	5925.6 / 70.3	OOM

Porous rocks dataset (Fig. 5, right, and Table 1) This dataset models porous material generated using various noise generators. The task consists of classifying according to the generator used. The task is rendered more difficult as the added ambient noise increases. Again, the Flood bifiltration outperforms the single-parameter methods when the level of noise is high ($p \geq 0.2$). Here, the number of landmarks has more impact on the accuracy score: this suggests that generator classification is easier when low-diameter 2-parameter persistence features are captured.

Time series datasets Finally, we conducted classification experiments on the time series UCR datasets [28] using time-delay embeddings. Table 2 (and Table 3 in Appendix D) show that, generally speaking, the sublevel Flood bifiltration is on par with the Delaunay bifiltration in terms of classification accuracy. Its key strength rather lies in enabling the use of larger embeddings (i.e., longer time series): see the out-of-memory (OOM) errors encountered with Delaunay on `KLC`, `CD`, `BH`. The inlined figure shows the classification time from the MMA (averaged over the cross-validation trials, and divided by the training dataset size) and illustrates the improved scalability of our approach.



5 Conclusion and future work

In this work, we introduced the *sublevel Flood bifiltration*, an extension of the Flood filtration to the context of sublevel-based multiparameter persistent homology. It provides a practical approximation of the reference sublevel Delaunay bifiltration while being more scalable.

Some limitations of this work are inherent to the landmark-based approach. Indeed, the construction will miss the topological features that are smaller than the resolution of $\mathrm{Del}(L)$. Besides, relying on $\mathrm{Del}(L)$ limits the ambient dimension that we are able to consider, and leads to instabilities when the

Table 2: Accuracy scores on UCR time series datasets. $\mathcal{D}$ and $\mathcal{F}$ denote the Delaunay and Flood bifiltrations, respectively.

Dataset	ED	DTWc	Čech	$\mathcal{D}$/MPL	$\mathcal{F}$/MPL	$\mathcal{D}$/SCDR	$\mathcal{F}$/SCDR
DPOAG	0.626	**0.770**	0.698	0.741	0.669	0.698	0.719
DPOC	0.717	0.717	0.699	0.641	0.649	0.659	**0.732**
DPTW	0.633	0.590	**0.647**	0.576	**0.647**	0.597	0.640
IPD	**0.955**	0.950	0.710	0.722	0.724	0.747	0.724
PC	**0.933**	0.878	0.889	0.744	0.728	0.883	0.911
LN2	0.754	**0.869**	0.721	0.525	0.541	0.607	0.705
Haptics	0.370	**0.377**	0.367	0.192	0.237	0.347	0.198
ACSF1	0.540	0.640	0.740	0.380	0.380	0.400	**0.750**
PigCVP	0.082	0.154	0.139	0.019	0.019	**0.226**	0.207
PArtP	0.125	0.245	0.361	0.029	0.024	**0.635**	0.433
Rock	**0.840**	0.600	0.420	0.420	0.420	0.420	0.420
KLC	0.278	0.815	**0.835**	OOM	0.145	OOM	0.802
CD	0.445	0.610	0.555	OOM	0.555	OOM	**0.640**
BH	**0.746**	0.610	0.737	OOM	0.717	OOM	0.737

combinatorics of $\mathrm{Del}(L)$ changes. As future work, we plan to investigate more deeply the choice of landmarks, for instance with density-aware approaches. Finally, we think that there remains room for improving the theoretical guarantees of both Flood filtration and sublevel Flood bifiltration, to explain, possibly with more assumptions, why this approach provides far better results than subsampling in many cases.

Acknowledgments

This work is partially supported by the European Commission grant ERC-2019-COG *"TORI"* (ref. 863464, `https://erc-tori.github.io/`) and by ANR-23-PEIA-0004 (PDE-AI, `https://pde-ai.math.cnrs.fr/`). We gratefully acknowledge support from the CNRS/IN2P3 Computing Center (Lyon, France) for providing computing and data-processing resources needed for this work.

References

[1] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631, 2019.

[2] Á. J. Alonso. A sparse multicover bifiltration of linear size. In *41st International Symposium on Computational Geometry (SoCG 2025)*, pages 6–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2025.

[3] Á. J. Alonso, M. Kerber, T. Lam, and M. Lesnick. Delaunay bifiltrations of functions on point clouds. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 4872–4891. SIAM, 2024.

[4] H. Asashiba, E. G. Escolar, K. Nakashima, and M. Yoshiwaki. On approximation of 2d persistence modules by interval-decomposables. *Journal of Computational Algebra*, 6:100007, 2023.

[5] U. Bauer. Ripser: efficient computation of Vietoris–Rips persistence barcodes. *Journal of Applied and Computational Topology*, 2021.

[6] U. Bauer, T. K. Dey, M. Kerber, F. Russold, and M. Söls. Fast free resolutions of bifiltered chain complexes. In *42nd International Symposium on Computational Geometry (SoCG 2026)*, pages 10:1–10:21. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2026.

[7] U. Bauer and H. Edelsbrunner. The Morse theory of Čech and Delaunay complexes. *Transactions of the American Mathematical Society*, 369(5):3741–3762, 2017.

[8] H. B. Bjerkevik, M. B. Botnan, and M. Kerber. Computing the interleaving distance is NP-hard. *Foundations of Computational Mathematics*, 20(5):1237–1271, 2020.

[9] H. B. Bjerkevik and M. Kerber. Asymptotic improvements on the exact matching distance for 2-parameter persistence. *Journal of Computational Geometry*, 14(1):309–342, 2023.

[10] N. Blaser, M. Brun, O. H. Gardaa, and L. M. Salbu. Core bifiltration. *Journal of Applied and Computational Topology*, 9(4):30, 2025.

[11] A. J. Blumberg and M. Lesnick. Stability of 2-parameter persistent homology. *Foundations of Computational Mathematics*, 24(2):385–427, 2024.

[12] J.-D. Boissonnat and S. Pritam. Edge collapse and persistence of flag complexes. In *36th International Symposium on Computational Geometry (SoCG 2020)*, pages 19–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020.

[13] M. Botnan and M. Lesnick. An introduction to multiparameter persistence. In *Representations of Algebras and Related Structures: International Conference on Representations of Algebras, ICRA 2020, 9–25 November 2020*, pages 77–150. EMS Press, 2023.

[14] M. B. Botnan, V. Lebovici, and S. Oudot. On rectangle-decomposable 2-parameter persistence modules. *Discrete & Computational Geometry*, 68(4):1078–1101, 2022.

[15] G. Carlsson and A. Zomorodian. The theory of multidimensional persistence. In *Proceedings of the twenty-third annual symposium on Computational geometry*, pages 184–193, 2007.

[16] M. Carrière and A. Blumberg. Multiparameter persistence image for topological machine learning. *Advances in Neural Information Processing Systems*, 33:22432–22444, 2020.

[17] A. Cerri, B. D. Fabio, M. Ferri, P. Frosini, and C. Landi. Betti numbers in multidimensional persistent homology are stable functions. *Mathematical Methods in the Applied Sciences*, 36(12):1543–1557, 2013.

[18] W. Chachólski, M. Scolamiero, and F. Vaccarino. Combinatorial presentation of multidimensional persistent homology. *Journal of Pure and Applied Algebra*, 221(5):1055–1075, 2017.

[19] F. Chazal, D. Cohen-Steiner, and Q. Mérigot. Geometric inference for probability measures. *Foundations of Computational Mathematics*, 11(6):733–751, 2011.

[20] F. Chazal, V. De Silva, M. Glisse, and S. Oudot. *The structure and stability of persistence modules*, volume 10. Springer, 2016.

[21] F. Chazal, B. Fasy, F. Lecci, B. Michel, A. Rinaldo, and L. Wasserman. Robust topological inference: Distance to a measure and kernel distance. *Journal of Machine Learning Research*, 18(159):1–40, 2018.

[22] F. Chazal and B. Michel. An introduction to topological data analysis: fundamental and practical aspects for data scientists. *Frontiers in artificial intelligence*, 4:667963, 2021.

[23] M. Clémot, J. Digne, and J. Tierny. Delaunay–Rips filtration: a study and an algorithm. *arXiv preprint arXiv:2512.17382*, 2025.

[24] P. Climaco and J. Garcke. Density-aware farthest point sampling. *Transactions on Machine Learning Research*, 2026.

[25] J. Cochoy and S. Oudot. Decomposition of exact pfd persistence bimodules. *Discrete & Computational Geometry*, 63(2):255–293, 2020.

[26] R. Corbet, U. Fugacci, M. Kerber, C. Landi, and B. Wang. A kernel for multi-parameter persistent homology. *Computers & Graphics: X*, 2:100005, 2019.

[27] R. Corbet, M. Kerber, M. Lesnick, and G. Osang. Computing the multicover bifiltration. *Discrete & Computational Geometry*, 70(2):376–405, 2023.

[28] H. A. Dau, A. Bagnall, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, and E. Keogh. The ucr time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6):1293–1305, 2019.

[29] H. Edelsbrunner and J. Harer. *Computational topology: an introduction*. American Mathematical Soc., 2010.

[30] H. Edelsbrunner and G. Osang. The multi-cover persistence of euclidean balls. *Discrete & Computational Geometry*, 65(4):1296–1313, 2021.

[31] A. Fay, I. García-Redondo, Q. Wang, H. Dubossarsky, and A. Monod. The shape of adversarial influence: Characterizing LLM latent spaces with persistent homology. In *International Conference on Machine Learning*. PMLR, 2026.

[32] Y. Gardinazzi, K. Viswanathan, G. Panerai, A. Ansuini, A. Cazzaniga, and M. Biagetti. Persistent topological features in large language models. In *International Conference on Machine Learning*, pages 18811–18830. PMLR, 2025.

[33] M. Glisse and S. Pritam. Swap, shift and trim to edge collapse a filtration. In *38th International Symposium on Computational Geometry (SoCG 2022)*, pages 44–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2022.

[34] J. T. Gostick, Z. A. Khan, T. G. Tranter, M. D. Kok, M. Agnaou, M. Sadeghi, and R. Jervis. Porespy: A Python toolkit for quantitative analysis of porous media images. *Journal of Open Source Software*, 4(37):1296, 2019.

[35] F. Graf, P. Pellizzoni, M. Uray, S. Huber, and R. Kwitt. The Flood Complex: Large-scale persistent homology on millions of points. In *NeurIPS*, 2025.

[36] F. Hensel, M. Moor, and B. Rieck. A survey of topological machine learning methods. *Frontiers in Artificial Intelligence*, 4:681108, 2021.

[37] D. S. Hochbaum and D. B. Shmoys. A best possible heuristic for the k-center problem. *Mathematics of operations research*, 10(2):180–184, 1985.

[38] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu. LightGBM: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

[39] B. Keller, M. Lesnick, and T. L. Willke. Persistent homology for virtual screening. 2018.

[40] M. Kerber, M. Lesnick, and S. Oudot. Exact computation of the matching distance on 2-parameter persistence modules. In *35th International Symposium on Computational Geometry (SoCG 2019)*, pages 46–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2019.

[41] M. Kerber and A. Nigmetov. Efficient approximation of the matching distance for 2-parameter persistence. In *36th International Symposium on Computational Geometry (SoCG 2020)*, pages 53–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020.

[42] M. Kerber and A. Rolle. Fast minimal presentations of bi-graded persistence modules. In *2021 Proceedings of the workshop on algorithm engineering and experiments (ALENEX)*, pages 207–220. SIAM, 2021.

[43] M. Kerber and F. Russold. Graphcode: Learning from multiparameter persistent homology using graph neural networks. *Advances in Neural Information Processing Systems*, 37:41103–41131, 2024.

[44] M. Lesnick. The theory of the interleaving distance on multidimensional persistence modules. *Foundations of Computational Mathematics*, 15(3):613–650, 2015.

[45] M. Lesnick and M. Wright. Interactive visualization of 2-d persistence modules. *arXiv preprint arXiv:1512.00180*, 2015.

[46] M. Lesnick and M. Wright. Computing minimal presentations and bigraded Betti numbers of 2-parameter persistent homology. *SIAM Journal on Applied Algebra and Geometry*, 2022.

[47] D. Loiseaux, M. Carrière, and A. Blumberg. A framework for fast and stable representations of multiparameter persistent homology decompositions. *Advances in neural information processing systems*, 36:35774–35798, 2023.

[48] D. Loiseaux, M. Carrière, and A. J. Blumberg. Multi-parameter module approximation: an efficient and interpretable invariant for multi-parameter persistence modules with guarantees. *Journal of Applied and Computational Topology*, 9(4):1–60, 2025.

[49] D. Loiseaux and H. Schreiber. Multipers: Multiparameter Persistence for Machine Learning. *Journal of Open Source Software*, 9(103):6773, Nov. 2024.

[50] C. Maria, J.-D. Boissonnat, M. Glisse, and M. Yvinec. The Gudhi library: Simplicial complexes and persistent homology. In *International congress on mathematical software*, pages 167–174. Springer, 2014.

[51] A. Mishra and F. C. Motta. Stability and machine learning applications of persistent homology using the Delaunay–Rips complex. *Frontiers in Applied Mathematics and Statistics*, 9:1179301, 2023.

[52] M. Moor, M. Horn, B. Rieck, and K. Borgwardt. Topological Autoencoders. In *International conference on machine learning*. PMLR, 2020.

[53] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Z. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *NeurIPS*, 2019.

[54] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *JMLR*, 2011.

[55] J. A. Perea and J. Harer. Sliding windows and persistence: An application of topological methods to signal analysis. *Foundations of computational mathematics*, 15(3):799–838, 2015.

[56] J. B. Pérez, S. Hauke, U. Lupo, M. Caorsi, and A. Dassatti. giotto-ph: a Python library for high-performance computation of persistent homology of Vietoris–Rips filtrations. *arXiv preprint arXiv:2107.05412*, 2021.

[57] J. Reininghaus, S. Huber, U. Bauer, and R. Kwitt. A stable multi-scale kernel for topological machine learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4741–4748, 2015.

[58] M. Royer, F. Chazal, C. Levrard, Y. Umeda, and Y. Ike. Atol: measure vectorization for automatic topologically-oriented learning. In *International conference on artificial intelligence and statistics*, pages 1000–1008. PMLR, 2021.

[59] D. R. Sheehy. Linear-size approximations to the Vietoris–Rips filtration. In *Proceedings of the twenty-eighth annual symposium on Computational geometry*, pages 239–248, 2012.

[60] F. Takens. Detecting strange attractors in turbulence. In *Dynamical Systems and Turbulence*, pages 366–381. Springer, 1980.

[61] I. Trofimov, D. Cherniavskii, E. Tulchinskii, N. Balabin, S. Barannikov, and E. Burnaev. Learning topology-preserving data representations. In *International Conference on Learning Representations*, 2023.

[62] O. Vipond. Multiparameter persistence landscapes. *Journal of Machine Learning Research*, 21(61):1–38, 2020.

[63] C. Zhang, H. Wan, X. Shen, and Z. Wu. PVT: Point-voxel transformer for point cloud learning. *International Journal of Intelligent Systems*, 37(12):11985–12008, 2022.

[64] S. Zhang, M. Xiao, and H. Wang. GPU-accelerated computation of Vietoris–Rips persistence barcodes. In *36th International Symposium on Computational Geometry (SoCG 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

Supplementary material

A Additional background

In this section, we provide formal definitions for the multiparameter persistence modules and the distances between them that are used in the paper. We write $\preceq$ for the partial order on $\mathbb{R}^n$, i.e., $\alpha \preceq \beta$ if and only if $\alpha_i \leq \beta_i$ for all $i \in [\![1, n]\!]$.

Multiparameter persistence module A *multiparameter persistence module* with n parameters is an indexed family of vector spaces $M = (M_\alpha)_{\alpha \in \mathbb{R}^n}$, with a family of *transition maps* $\varphi_{\alpha,\beta} : M_\alpha \to M_\beta$ for all $\alpha \preceq \beta$, that satisfy $\varphi_{\alpha,\gamma} = \varphi_{\beta,\gamma} \circ \varphi_{\alpha,\beta}$ for all $\alpha \preceq \beta \preceq \gamma$.

Often, a multiparameter persistence module is obtained by applying the homology functor H_k on a multifiltration $(\mathcal{K}_\alpha)_{\alpha \in \mathbb{R}^n}$, giving $M = (H_k(\mathcal{K}_\alpha))_{\alpha \in \mathbb{R}^n}$. The linear maps $\varphi_{\alpha,\beta}$ are induced by the simplicial inclusions $\mathcal{K}_\alpha \subseteq \mathcal{K}_\beta$ for all $\alpha \preceq \beta$.

Interleaving distance The *interleaving distance* [44] allows multiparameter persistence modules to be compared. If M is a multiparameter persistence module, its ε-shift is the module $M_\varepsilon = (M_{\alpha+\vec{\varepsilon}})_{\alpha \in \mathbb{R}^n}$ with $\vec{\varepsilon} = (\varepsilon, \dots, \varepsilon) \in \mathbb{R}^n$. Two multiparameter persistence modules M and N (of transition maps φ and ψ respectively) are ε-interleaved if there exist $f : M \to N_\varepsilon$ and $g : N \to M_\varepsilon$ such that $g_{\cdot+\varepsilon} \circ f_\cdot = \varphi_{\cdot,\cdot+2\varepsilon}$ and $f_{\cdot+\varepsilon} \circ g_\cdot = \psi_{\cdot,\cdot+2\varepsilon}$. Then, the interleaving distance between two multiparameter persistence modules M and N is defined as

$$\mathrm{d_I}(M, N) = \inf\{\varepsilon > 0 \mid M \text{ and } N \text{ are } \varepsilon\text{-interleaved}\}.$$

Matching distance Computing the interleaving distance (and even any approximation by a factor smaller than 3) is NP-hard [8]. The *matching distance* [17] between two 2-parameter persistence modules M and N is defined over the associated fibered barcodes, i.e., the collection of (single-parameter) persistence barcodes obtained when restricting a module to one-dimensional slices ℓ defined by a line of equation $y = ax + b$ (giving the single-parameter persistence module $M_\ell = (M_{x,ax+b})_{x>0}$). For each slice ℓ, we define the following distance along ℓ:

$$\mathrm{d}^{M,N}_{\mathrm{slice}}(\ell) = w(\ell)\, \mathrm{d_b}(M_\ell, N_\ell) \text{ with } w(\ell) = \begin{cases} \frac{1}{\sqrt{1+a^2}} & \text{if } a \geq 1 \\ \frac{1}{\sqrt{1+1/a^2}} & \text{if } 0 < a < 1 \end{cases} \tag{8}$$

where $\mathrm{d_b}$ is the single-parameter bottleneck distance. Then, the matching distance is defined as

$$\mathrm{d_{match}}(M, N) = \sup_\ell \mathrm{d}^{M,N}_{\mathrm{slice}}(\ell) \tag{9}$$

where the sup is over the slices with $a > 0$. The correcting weights $w(\ell)$ are here to ensure that $\mathrm{d_{match}}(M, N) \leq \mathrm{d_I}(M, N)$. Both exact [40, 9] and approximate [41] algorithms have been proposed to compute the matching distance.

B Proofs

This section contains the proofs of the theorems stated in Sections 3.2 and 3.3.

Proof of Theorem 1. Let $\sigma \in \mathcal{F}_{r,s}(X, L, \gamma_X)$ and $p \in \operatorname{conv}\sigma$. There is an $x \in X$ such that $\mathrm{d}(p, x) \leq r$ and $\gamma_X(x) \leq s$. Then, there exists $y \in Y$ such that $\mathrm{d}(x, y) \leq \mathrm{d_H}(X, Y)$. Therefore $\mathrm{d}(p, y) \leq \mathrm{d}(p, x) + \mathrm{d}(x, y) \leq r + \mathrm{d_H}(X, Y)$. Besides:

$$\begin{aligned} \gamma_Y(y) - s &\leq \gamma_Y(y) - \gamma_X(x) \\ &\leq |\gamma_Y(y) - \gamma_X(x)| \\ &\leq |\gamma_Y(y) - \gamma_Y(x)| + |\gamma_Y(x) - \gamma_X(x)| \\ &\leq \ell\, \mathrm{d_H}(X, Y) + \|\gamma_X - \gamma_Y\|_\infty \end{aligned}$$

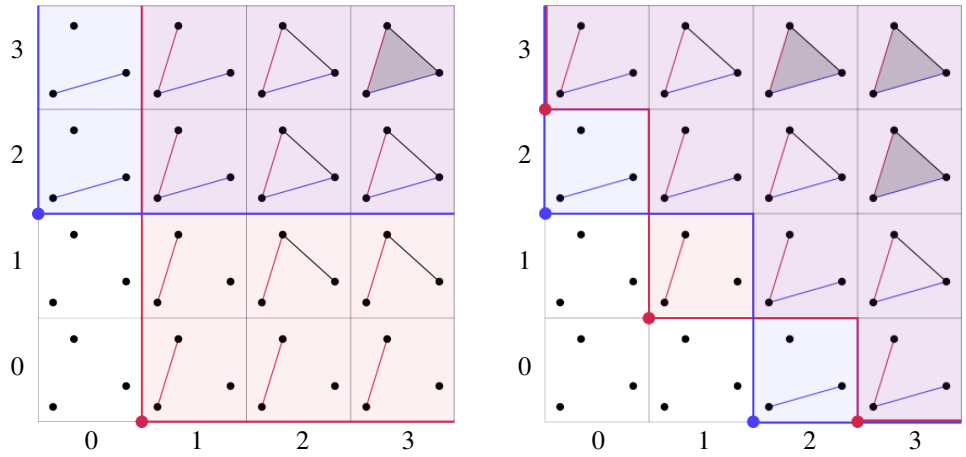


Figure 6: Two bifiltrations over a filled triangle indexed over $[\![0,3]\!]^2$. The left one is 1-critical while the right one is 3-critical. The blue (resp. red) dots represent the minimal set of bigrades $\phi(\sigma)$ of the blue (resp. red) edge. Their respective supports $\mathrm{supp}(\sigma)$ are highlighted with the same color.

This means that $\sigma \in \mathcal{F}_{r+\mathrm{d_H}(X,Y),s+\ell\,\mathrm{d_H}(X,Y)+\|\gamma_X-\gamma_Y\|_\infty}(Y,L,\gamma_Y)$. By writing $\alpha = \max\left(\mathrm{d_H}(X,Y), \ell\,\mathrm{d_H}(X,Y)+\|\gamma_X-\gamma_Y\|_\infty\right)$, we have:

$$\mathcal{F}_{r,s}(X,L,\gamma_X) \subseteq \mathcal{F}_{r+\alpha,s+\alpha}(Y,L,\gamma_Y)$$
$$\mathcal{F}_{r,s}(Y,L,\gamma_Y) \subseteq \mathcal{F}_{r+\alpha,s+\alpha}(X,L,\gamma_X)$$

These simplicial inclusions provide an α-interleaving between the two modules $\mathbb{F}(X,L,\gamma_X)$ and $\mathbb{F}(Y,L,\gamma_Y)$, hence the result. □

Proof of Theorem 2. Under these conditions we have $\mathrm{d_H}(X,Y) \leq \varepsilon$. Besides:

- if γ is obtained by KDE with an ℓ_K-Lipschitz kernel K and bandwidth h, then both γ_X and γ_Y are $\frac{\ell_K}{h^{d+1}}$-Lipschitz and one easily checks that $\|\gamma_X-\gamma_Y\|_\infty \leq \varepsilon\frac{\ell_K}{h^{d+1}}$;
- if γ is obtained by DTM with parameter $m \in (0,1]$, then both γ_X and γ_Y are 1-Lipschitz, and besides (see, e.g., Theorem 3 of [21] for the properties of DTM):

$$\|\gamma_X-\gamma_Y\|_\infty \leq \frac{1}{\sqrt{m}}\mathcal{W}_2(X,Y) \leq \varepsilon\frac{1}{\sqrt{m}}.$$

Theorem 1 allows us to obtain the upper bound on the interleaving distance between both modules in the two cases. □

Proof of Theorem 3. Recall that $\widetilde{\mathcal{F}}_{r,s}(X,L,\gamma) = \left\{\sigma \in \mathrm{Del}(L) \mid \mathrm{conv}(\sigma) \subset \bigcup_{x\in X_s^\sigma} B_r(x)\right\}$ with

$$X^\sigma = \bigcup_{\tau\in\mathrm{C}(\sigma)} \left(X \cap B_{\sqrt{2}r_\tau}(c_\tau)\right) \cup \bigcup_{l\in\mathrm{V}(\sigma)} X^{\{l\}} \quad \text{and} \quad X_s^\sigma = \{x \in X^\sigma \mid \gamma(x) \leq s\}.$$

We first show that $\widetilde{\mathcal{F}}_{r,s}(X,L,\gamma)$ is a simplicial complex. Let $\sigma \in \widetilde{\mathcal{F}}_{r,s}(X,L,\gamma)$ and $\tau \subset \sigma$. We have $\mathrm{C}(\sigma) \subset \mathrm{C}(\tau)$ and $\mathrm{V}(\sigma) \subset \mathrm{V}(\tau)$ because the cofaces of σ are also cofaces of τ. Therefore, $X^\sigma \subset X^\tau$, hence $X_s^\sigma \subset X_s^\tau$. This implies that $\mathrm{conv}(\tau) \subset \mathrm{conv}(\sigma) \subset \bigcup_{x\in X_s^\sigma} B_r(x) \subset \bigcup_{x\in X_s^\tau} B_r(x)$, hence $\tau \in \widetilde{\mathcal{F}}_{r,s}(X,L,\gamma)$. In addition, $(\widetilde{\mathcal{F}}_{r,s}(X,L,\gamma))_{(r,s)\in\mathbb{R}^2}$ is clearly increasing with respect to the partial order $\preceq$ in $\mathbb{R}^2$. It is therefore a bifiltration.

As $X_s^\sigma \subset X_s$ for all s and σ, we have $\widetilde{\mathcal{F}}_{r,s}(X,L,\gamma) \subseteq \mathcal{F}_{r,s}(X,L,\gamma)$. Conversely, let $\sigma \in \mathcal{F}_{r,s}(X,L,\gamma)$ and $p \in \mathrm{conv}\,\sigma$. There is an $x \in X$ such that $\mathrm{d}(p,x) \leq r$ and $\gamma(x) \leq s$. If $x \in X^\sigma$,

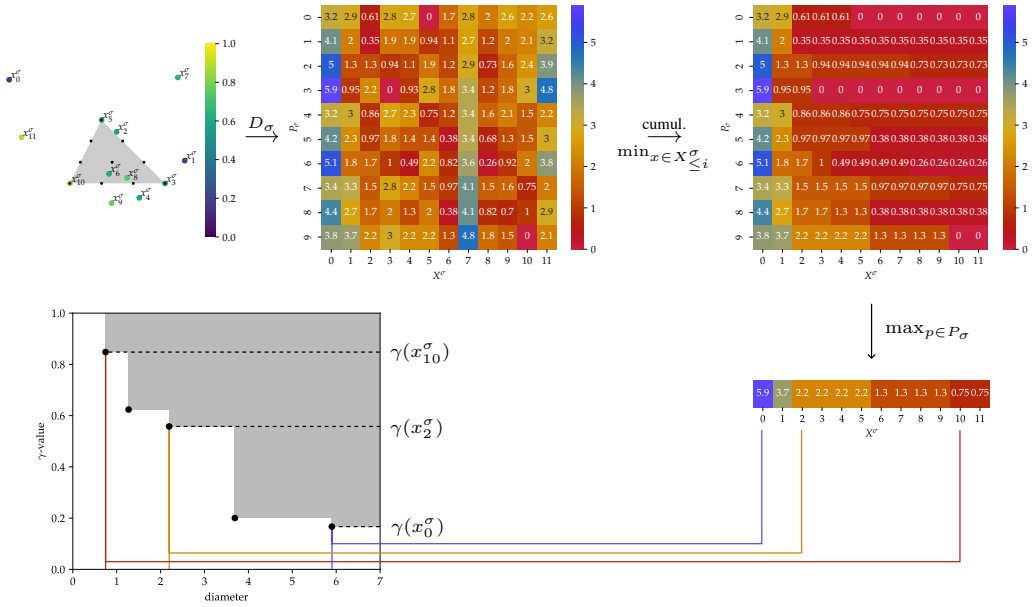


Figure 7: Illustration of the tensor operations that compute the set of bigrades of a triangle $\sigma \in \mathrm{Del}(L)$. **Top left:** the sample P_σ (black dots) has size 10 and the set of points $X^\sigma = \{x^\sigma_0, \dots, x^\sigma_{11}\}$ has size 12 and is sorted and colored by γ value. **Bottom left:** the support of σ (in gray, with the minimal set of bigrades $\phi(\sigma)$ shown with black dots) is computed from the distance matrix $D_\sigma = \mathrm{d}(P_\sigma, X^\sigma)$ to which we apply a cumulative minimum along each row, and then a maximum along each column.

then we directly have $p \in \bigcup_{x \in X^\sigma_s} B_r(x)$. If $x \notin X^\sigma$, this means in particular that for all $l \in \sigma$, $x \notin X^{\{l\}}$. Recall that $X^{\{l\}}$ contains the points that contribute to the minimal set of bigrades of $\{l\}$. Therefore, for each $l \in \sigma$, there exists an $x_l \in X^{\{l\}} \subset X^\sigma$ such that $\mathrm{d}(x_l, l) \leq \mathrm{d}(x, l)$ and $\gamma(x_l) \leq \gamma(x) \leq s$ (hence $x_l \in X^\sigma_s$). We have:

$$\begin{aligned}
\mathrm{d}(p, X^\sigma_s) &\leq \min_{l \in \sigma} \mathrm{d}(p, x_l) \\
&\leq \min_{l \in \sigma} \left(\mathrm{d}(p, l) + \mathrm{d}(x_l, l)\right) \\
&\leq \min_{l \in \sigma} \left(\mathrm{d}(p, l) + \mathrm{d}(x, l)\right) \\
&\leq \min_{l \in \sigma} \left(\mathrm{d}(p, l) + \mathrm{d}(x, p) + \mathrm{d}(p, l)\right) \\
&\leq 2 \min_{l \in \sigma} \left(\mathrm{d}(p, l)\right) + r \\
&\leq 2\Delta(L) + r
\end{aligned}$$

Hence $p \in \bigcup_{x \in X^\sigma_s} B_{r+2\Delta(L)}(x)$. Therefore, $\mathcal{F}_{r,s}(X, L, \gamma) \subseteq \widetilde{\mathcal{F}}_{r+2\Delta(L),s}(X, L, \gamma)$. The two inclusions show that the multiparameter persistence modules of $\mathcal{F}(X, L, \gamma)$ and $\widetilde{\mathcal{F}}(X, L, \gamma)$ are $2\Delta(L)$-interleaved, hence the result. □

C Computational approach

Fig. 7 illustrates the batched tensor operations we use to compute the set of bigrades of a simplex in the sublevel Flood bifiltration. Given the distance matrix $D_\sigma = \mathrm{d}(P_\sigma, X^\sigma)$, we obtain the (non-minimal) set of bigrades by first applying a cumulative minimum operation along each row, and then taking the maximum on each column.

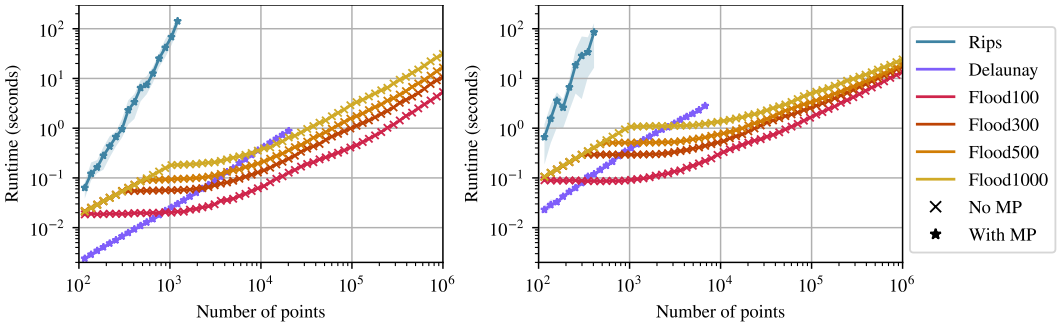


Figure 8: Running time of the bifiltration computation (plus its minimal presentation for Rips and Delaunay) for $X \subset \mathbb{R}^2$ (left) and $X \subset \mathbb{R}^3$ (right). Contrary to Fig. 3, this does not include multiparameter module approximation.

D Additional experiments

This section contains additional experiments that we did not include in the main paper.

D.1 Runtimes

Fig. 8 shows the computation runtime of several bifiltrations, without including the multiparameter module approximation (MMA) (unlike Fig. 3 which includes it).

D.2 Structure sizes

Fig. 9 shows the size of the bifiltrations (in terms of total number of bigrades) and the size of their multiparameter module approximation (in terms of number of summands in the module decomposition). It also shows how the average criticality of the simplices in the sublevel Flood bifiltration increases with the number of points and decreases with the number of landmarks (sublevel Rips and Delaunay bifiltrations are not represented as their criticality is always 1).

D.3 Empirical stability evaluation

Fig. 10 shows a study of the stability of the Flood bifiltration with respect to the input X, when L is fixed (as in Theorems 1 and 2) and when L is obtained separately from X and its perturbation.

D.4 Additional classification results

Fig. 11 is similar to Fig. 5, left, but with 20,000 points in $\mathbb{R}^2$ and 5,000 points in $\mathbb{R}^3$ per point set. Table 3 gives the accuracy scores on the UCR datasets that are not shown in the main paper.

D.5 Landmark sampling evaluation

We compare several sampling techniques to obtain a set of landmarks $L \subseteq X$ of prescribed size: furthest point sampling (FPS), random subsampling, and the density-aware approach (DA-FPS) proposed in [24] (see Fig. 12 for an illustration of these three approaches). Table 4 evaluates those approaches by showing the associated matching distance to the reference sublevel Delaunay bifiltration on the noisy ring (in $\mathbb{R}^2$) and noisy sphere (in $\mathbb{R}^3$).

D.6 Number of landmarks

To better illustrate how the number of landmarks has an influence in practice, we show how the classification accuracy score evolves when the number of landmarks increases for the toy synthetic dataset (in $\mathbb{R}^2$) in Fig. 13, left, and for the porous rock dataset (in $\mathbb{R}^3$) in Fig. 13, right. Additionally, we use in Fig. 14 a variant of the 2-dimensional toy synthetic dataset, with two classes whose point sets feature respectively 124 and 125 non-intersecting empty regions, each of radius 0.05, and placed

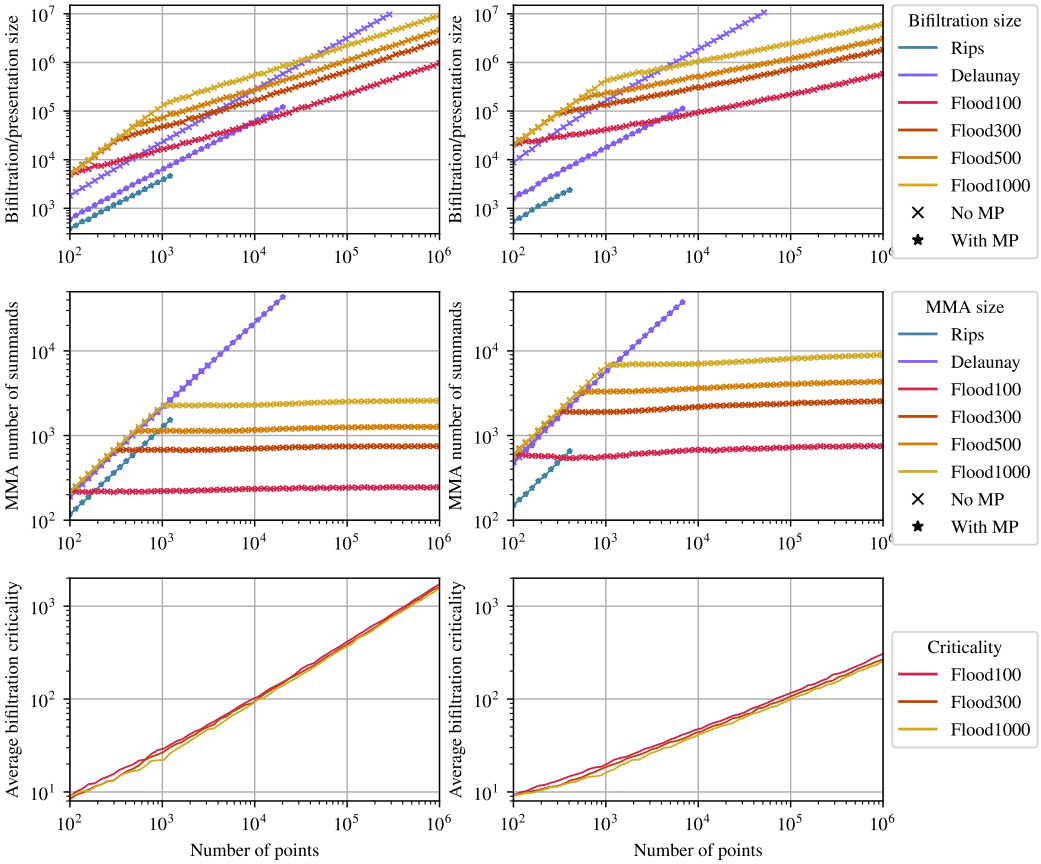


Figure 9: **Top row:** size of the bifiltrations, i.e., total number of stored bigrades. For 1-critical bifiltrations (all except sublevel Flood), this corresponds to the number of simplices. For the minimal presentations, the total number of generators is shown instead (all homology dimensions). **Middle row:** total number of summands in the multiparameter module approximation (all homology dimensions). **Bottom row:** average criticality $|\phi(\sigma)|$ in the sublevel Flood bifiltrations.

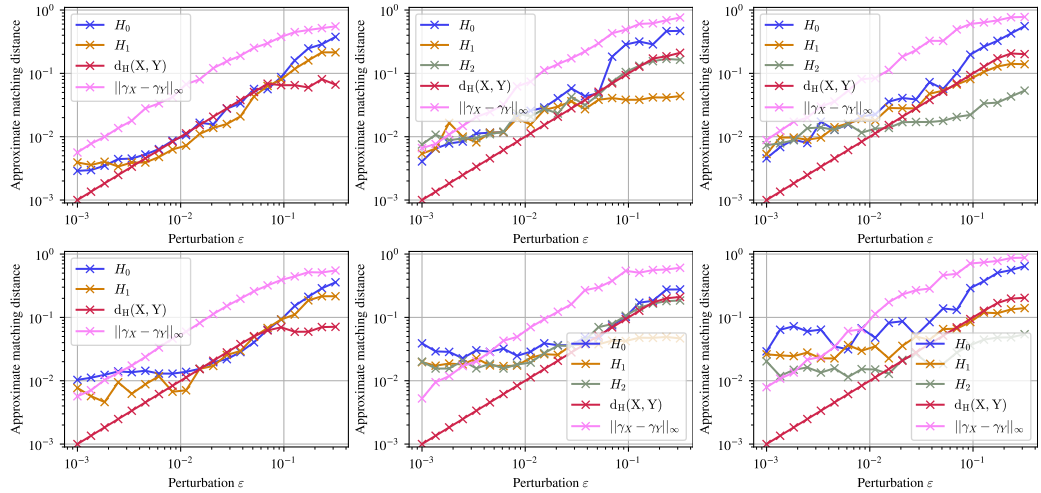


Figure 10: Stability of the sublevel Flood bifiltration with respect to X, as measured by the matching distance between $\mathbb{F}(X, L, \gamma_X)$ and $\mathbb{F}(X', L', \gamma_{X'})$ with $X' = X + E$ and with γ_X and $\gamma_{X'}$ obtained as KDEs. In the top row, $L = L'$ is obtained as an FPS on X (fixed landmarks, as in Theorems 1 and 2). In the bottom row, L' is computed as an FPS from X'. L has 1,000 points and X has 10,000 points distributed over the noisy ring in $\mathbb{R}^2$ (left), the noisy sphere in $\mathbb{R}^3$ (center) and the noisy torus in $\mathbb{R}^3$ (right).

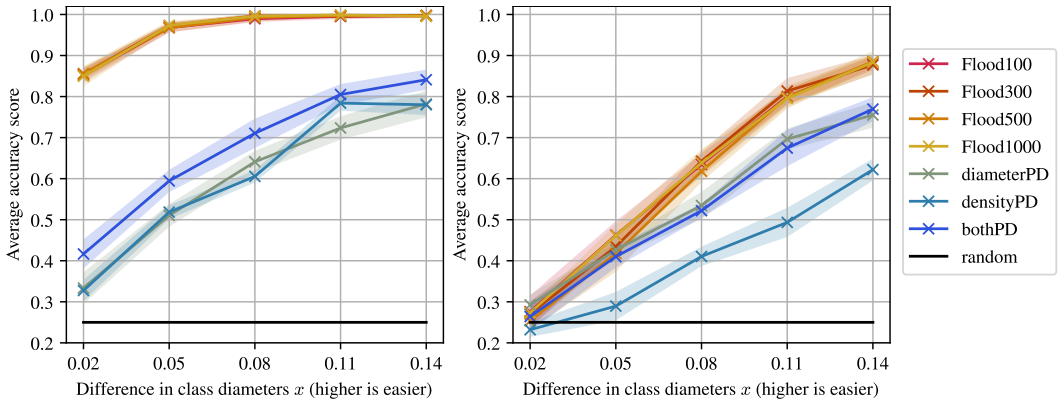
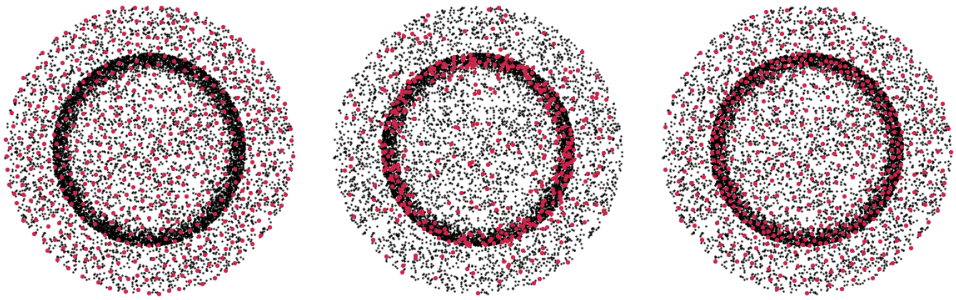


Figure 11: Accuracy scores (averaged over 10 splits) of single-parameter PH (diameterPD, densityPD and bothPD) and multiparameter PH (Flood) on the synthetic noisy swisscheese toy datasets parameterized by difficulty x. Here, the dataset contains 800 sets equally split into four classes characterized by the diameter of their low-density regions. The black lines show the score of a random classifier. **Left:** Each point set contains 20,000 points in $\mathbb{R}^2$. **Right:** Each point set contains 5,000 points in $\mathbb{R}^3$.

Table 3: Accuracy scores on additional UCR time series datasets. $\mathcal{D}$ and $\mathcal{F}$ denote the Delaunay and Flood bifiltrations, respectively.

Dataset	ED	DTWc	Čech	$\mathcal{D}$/MPL	$\mathcal{F}$/MPL	$\mathcal{D}$/SCDR	$\mathcal{F}$/SCDR
PPOAG	0.785	0.805	0.785	**0.829**	0.785	0.815	0.815
PPOC	0.808	0.783	**0.832**	0.663	0.674	0.814	0.821
PPTW	0.707	0.756	0.737	**0.776**	0.756	0.741	0.771
ECG200	**0.880**	0.770	0.710	0.740	0.790	0.830	0.830
MI	0.684	**0.737**	0.584	0.533	0.547	0.628	0.514
Plane	0.962	**1.000**	0.876	0.629	0.790	0.933	0.610
SL	0.789	0.792	0.738	0.656	0.552	0.842	**0.864**
GP	0.913	0.907	0.880	0.767	0.673	0.940	**0.960**
GPAS	0.899	0.918	**0.949**	0.522	0.494	0.940	0.937
GPMVF	0.975	**0.997**	0.953	0.842	0.883	0.972	0.962
GPOVY	0.952	0.838	**1.000**	0.994	0.625	0.590	**1.000**
SC	0.880	**0.993**	0.503	0.547	0.550	0.580	0.593
EQ	0.712	0.719	**0.763**	0.647	0.705	0.755	0.741
PAirP	0.058	0.106	**0.264**	0.043	0.024	0.077	0.048
HT	0.664	**0.924**	0.815	0.672	0.580	0.395	0.807



FPS random DA-FPS

Figure 12: Three sampling approaches to get 500 landmarks (red dots) from an input $X \subset \mathbb{R}^2$ of size 10,000 (black dots).

Table 4: Matching distance between the sublevel Delaunay and Flood bifiltrations for several landmark sampling techniques: classical and density-aware furthest point sampling, and random sampling. The input point sets have 10,000 points and 500 landmarks are selected. The matching distances are averaged over 100 runs.

	Noisy ring ($\mathbb{R}^2$)		Noisy sphere ($\mathbb{R}^3$)		
	$\mathrm{d}_{\mathrm{match}}$ (H_0)	$\mathrm{d}_{\mathrm{match}}$ (H_1)	$\mathrm{d}_{\mathrm{match}}$ (H_0)	$\mathrm{d}_{\mathrm{match}}$ (H_1)	$\mathrm{d}_{\mathrm{match}}$ (H_2)
FPS	0.022 ± 0.005	0.020 ± 0.005	0.059 ± 0.011	0.052 ± 0.005	0.067 ± 0.009
random	0.012 ± 0.004	0.012 ± 0.004	0.044 ± 0.007	0.032 ± 0.011	0.045 ± 0.017
DA-FPS	0.012 ± 0.003	0.009 ± 0.003	0.049 ± 0.008	0.035 ± 0.004	0.038 ± 0.007

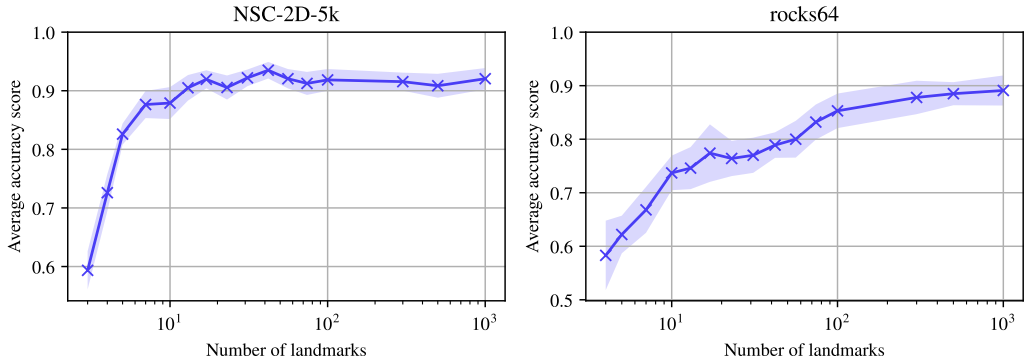


Figure 13: Accuracy score for two classification problems using the sublevel Flood bifiltration, as a function of the number of landmarks.

randomly. The classification task involves determining which of these two classes the point set belongs to.

The three datasets exhibit different behaviors. For the toy synthetic dataset, few landmarks are required to obtain a good classification accuracy because low-density regions do not have to be captured individually to retrieve their diameter. For the porous rock dataset, the classification accuracy increases more regularly with the number of landmarks, since smaller and smaller geometric details are captured in the point sets. Finally, for the dataset depicted in Fig. 14, there is a sharp transition between low and high accuracy, because classifying correctly requires counting precisely the number of voids and therefore capturing them separately, which happens after a threshold number of landmarks is reached.

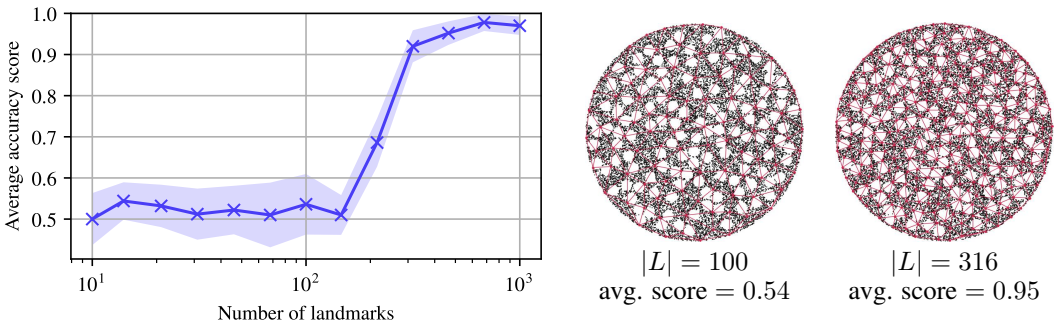


Figure 14: Accuracy score for the classification problem involving counting the number of voids, using the sublevel Flood bifiltration, as a function of the number of landmarks. On the right, we show the Delaunay triangulation of the set of landmarks when the classification is unsuccessful ($|L| = 100$) or successful ($|L| = 316$).

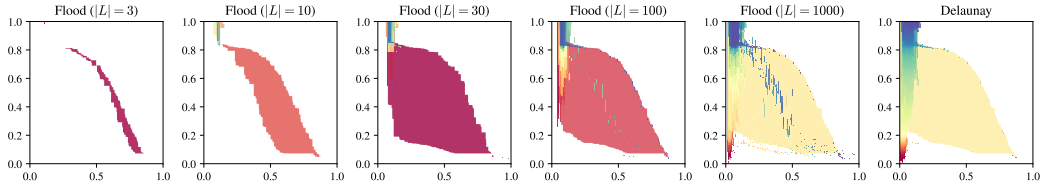


Figure 15: Multiparameter module approximation when the number of landmarks increases, for a point set X of size 2,000 sampled as a noisy ring in $\mathbb{R}^2$.

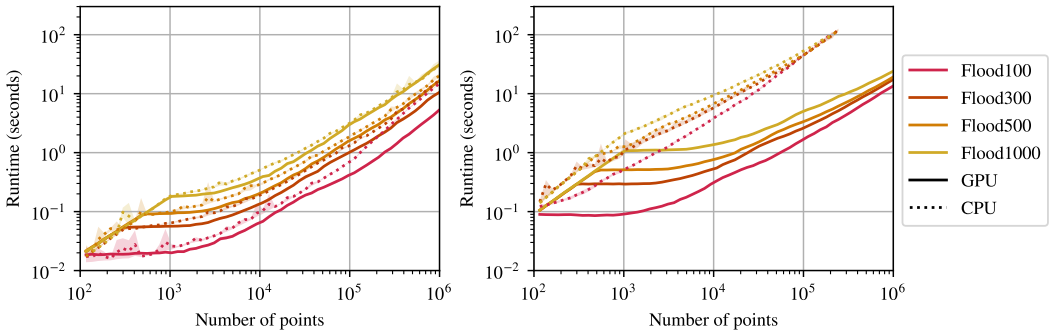


Figure 16: Comparison of runtimes for computing the sublevel Flood bifiltration (no MMA) on a GPU and a CPU (we used a Core i7-13850HX CPU for this experiment).

Finally, to provide a more visual insight into the influence of the number of landmarks, Fig. 15 shows the multiparameter module approximation of the sublevel Flood bifiltration as the number of landmarks increases and the sublevel Delaunay bifiltration is increasingly well approximated.

D.7 Interest of GPU use

In order to clarify how the GPU batching helps to speed up the computation of the set of bigrades of the simplices, we show in Fig. 16 the runtime comparison (bifiltration only, no MMA) between GPU (left number) and CPU (right number) on the 2-dimensional noisy ring and 3-dimensional noisy sphere datasets. For this experiment, we used a Core i7-13850HX CPU (with 20 physical cores). It illustrates that the use of the GPU is significantly profitable, especially starting from $d = 3$. This is because the higher number of samples (1540 samples for a tetrahedron against 210 for a triangle, with 20 samples per edge) makes the GPU batching more efficient. Broadly speaking, our approach trades the costly PH computation on a large filtration for the costly but GPU-parallelizable construction of a much smaller one, on which PH is then much faster to compute.

D.8 Masking strategy empirical evaluation

Table 5 provides an empirical evaluation of the effect of the masking strategy described in Section 3.3 on the approximation (of the reference sublevel Delaunay bifiltration) and the runtime. A general observation is that—as expected—the exact version (i.e., taking the whole X instead of X^σ into account for all simplices σ) is significantly costlier in runtime and memory. In addition, empirically, the loss of precision of the approximate version compared to the exact one tends to be less significant when the size of X increases.

D.9 Codensity hyperparameter

The choice of the parameters of the estimators (e.g., bandwidth h for KDE, m for DTM) influences the accuracy in classification tasks, similarly to other methods. Table 6 shows the accuracy obtained with the sublevel Flood (300 landmarks) and Delaunay bifiltrations on the 2-dimensional noisy swisscheese dataset with $x = 0.05$ (see Fig. 5, left) for Gaussian kernel density estimation with

Table 5: Comparison between the exact (E) and masked (M) approaches (see Section 3.3) in terms of runtime and approximation of the sublevel Delaunay bifiltration.

$\lvert X\rvert$	$\lvert L\rvert$	Runtime (s)		$\mathrm{d}_{\mathrm{match}}$ (H_0)		$\mathrm{d}_{\mathrm{match}}$ (H_1)		$\mathrm{d}_{\mathrm{match}}$ (H_2)	
		M	E	M	E	M	E	M	E
1000	100	0.14	0.39	0.17	0.12	0.04	0.05	0.05	0.02
	300	0.46	1.38	0.09	0.09	0.03	0.03	0.06	0.01
	500	0.76	2.46	0.07	0.07	0.03	0.03	0.06	0.01
	1000	1.70	5.07	0.01	0.01	0.03	0.03	0.02	0.01
20000	100	0.55	5.03	0.05	0.05	0.08	0.10	0.10	0.11
	300	1.12	18.85	0.04	0.05	0.07	0.06	0.10	0.08
	500	1.65	33.36	0.03	0.04	0.06	0.05	0.08	0.07
	1000	2.99	70.52	0.03	0.03	0.05	0.05	0.05	0.05

Table 6: Average classification accuracy on the 2-dimensional swisscheese dataset with the sublevel Flood (300 landmarks) and Delaunay bifiltrations, using Gaussian KDE with several bandwidths h.

h	Flood (300 l.)	Delaunay
0.004	0.793 ± 0.027	0.817 ± 0.019
0.01	0.891 ± 0.016	0.920 ± 0.017
0.04	0.916 ± 0.015	0.923 ± 0.017
0.1	0.774 ± 0.029	0.794 ± 0.025
0.4	0.429 ± 0.035	0.454 ± 0.031

several bandwidths. We show the average over 10 splits. Fig. 5 used the same value 0.04 for all the methods.

E Data description

This section contains a description of the synthetic and real datasets used throughout the paper.

Toy synthetic dataset Our first test case is a synthetic toy dataset, `noisy_swiss_cheese`, that aims to illustrate the typical case where multiparameter PH carries topological information that is not faithfully detected by either diameter-based or density-based single-parameter PH, or even by the combination of the two. This dataset is made of points in the unit ball (in $\mathbb{R}^2$ and $\mathbb{R}^3$), uniformly sampled, except in a variable number $k \in [\![1, 10]\!]$ of spherical regions that have a lower density. For a given point set, these lower-density regions have the same diameter δ among $\{\delta_0, \delta_0 + x/3, \delta_0 + 2x/3, \delta_0 + x\}$, with $\delta_0 = 0.1$ and $x > 0$. The classification task involves determining which of the four diameter classes the point set belongs to. The parameter x quantifies the difference in class diameters and the problem's difficulty: as x increases, the classes become easier to separate. We only use H_1 (for $\mathbb{R}^2$) or H_2 (for $\mathbb{R}^3$) for all the methods. Results are shown in Fig. 5, left.

Porous rocks dataset The second dataset models porous material and is generated with the `PoreSpy` library [34]. It generates binary 64^3 grids that we convert into a point set, with some Gaussian noise (to avoid the non-generic position due to the lattice structure). The porosity is chosen randomly, uniformly in $[0.05, 0.95]$. The classification task consists of determining which porous material generator was used, between: `blobs` that generates random noise combined with Gaussian blur, with a blobiness parameter chosen uniformly in $[0.5, 2]$; and `fractals` which is simplex noise with a frequency chosen uniformly in $[0.05, 0.2]$. To measure the ability of multiparameter PH to handle noise and outliers, we parameterize the difficulty of the problem with an ambient noise parameter which represents the probability p of each element of the binary grid to be set to 1 (after the binary generation and before the point set conversion). We only use H_2. Results are shown in Fig. 5, right.

Time series datasets We perform experiments involving time series classification. TDA has been applied to such problems [55] by applying PH to the time-delay Takens embeddings [60] of the time series. Indeed, signal properties (e.g., periodicity) may translate into specific topological features in the embedding (e.g., loops). Multiparameter PH was also recently used in this context (see, e.g.,

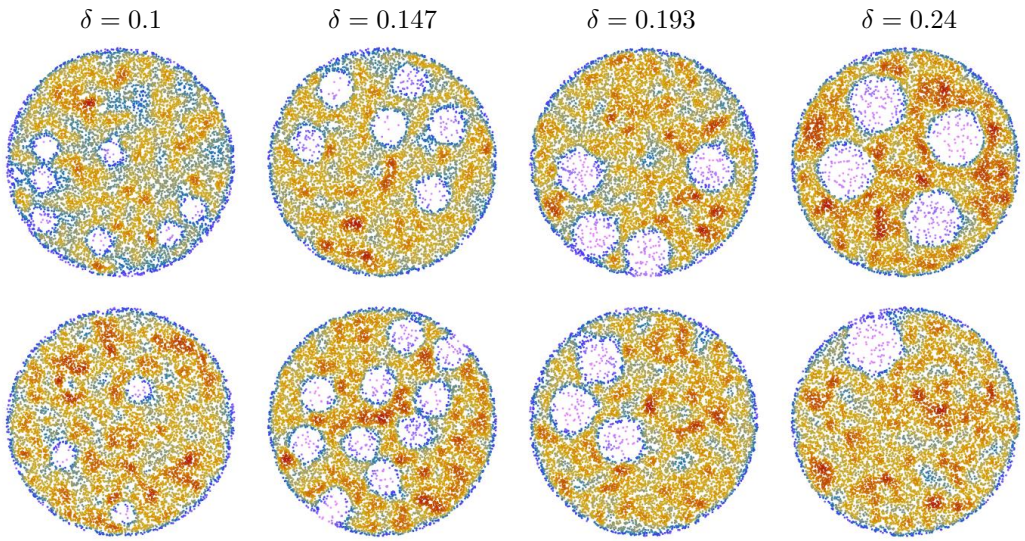


Figure 17: Two examples of each class in the `noisy_swiss_cheese` dataset in $\mathbb{R}^2$, with 10,000 points in each point set and difficulty parameter $x = 0.14$.

[16, 47]), with the idea that noise in the time series transmits to the embedding. We take several time series datasets from the UCR archive [28] of various sizes, lengths, numbers of classes and coming from different scientific fields (see Table 7 for the description of the datasets used) and use the suggested train and test datasets. We encode them into $\mathbb{R}^3$, with time-delay Takens embedding, and apply the same pipelines as above, considering H_0 and H_1, with a DTM with parameter $m = 0.1$ as codensity estimation. Results are shown in Tables 2 and 3, in which we also report the baseline scores provided by [28] (except for `KLC`, `CD` and `BH`, which we ran ourselves as they do not belong to the original UCR dataset): nearest neighbor with Euclidean distance (ED), and dynamic time warping with constant window width (DTWc).

Table 7: The list of the UCR datasets used in this paper, with their acronym and their full name, the size of train and test sets, the length of the time series and the number of classes.

Dataset	UCR dataset full name	Train	Test	Length	Classes	Category
DPOAG	DistalPhalanxOutlineAgeGroup	400	139	80	3	image
DPOC	DistalPhalanxOutlineCorrect	600	276	80	2	image
DPTW	DistalPhalanxTW	400	139	80	6	image
PPOAG	ProximalPhalanxOutlineAgeGroup	400	205	80	3	image
PPOC	ProximalPhalanxOutlineCorrect	600	291	80	2	image
PPTW	ProximalPhalanxTW	400	205	80	6	image
ECG200	ECG200	100	100	96	2	ecg
IPD	ItalyPowerDemand	67	1029	24	2	sensor
MI	MedicalImages	381	760	99	10	image
Plane	Plane	105	105	144	7	sensor
SL	SwedishLeaf	500	625	128	15	image
GP	GunPoint	50	150	150	2	har
GPAS	GunPointAgeSpan	135	316	150	2	har
GPMVF	GunPointMaleVersusFemale	135	316	150	2	har
GPOVY	GunPointOldVersusYoung	136	315	150	2	har
PC	PowerCons	180	180	144	2	device
SC	SyntheticControl	300	300	60	6	simulated
EQ	Earthquakes	322	139	512	2	sensor
LN2	Lightning2	60	61	637	2	sensor
Haptics	Haptics	155	308	1092	5	motion
ACSF1	ACSF1	100	100	1460	10	device
PigCVP	PigCVP	104	208	2000	52	hemodynamics
PArtP	PigArtPressure	104	208	2000	52	hemodynamics
PAirP	PigAirwayPressure	104	208	2000	52	hemodynamics
HT	HouseTwenty	40	119	2000	2	device
Rock	Rock	20	50	2844	4	spectro
KLC	KeplerLightCurves	920	399	4767	7	sensor
CD	CatsDogs	164	164	14773	2	audio
BH	BinaryHeartbeat	204	205	18530	2	audio

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We estimate that our abstract and introduction provide a clear description of the claimed contributions: theoretical properties (Section 3.2), algorithm (Section 3.3), and experimental validation (Section 4).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in the conclusion (Section 5).

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The proofs of Theorems 1 and 2 are provided in Appendix B.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Our algorithm is described in Section 3.3. Our experimental setup is specified at the beginning of Section 4, while our classification datasets are specified in Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Code will be made publicly available on GitHub in case of acceptance.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Our experimental setup is specified at the beginning of Section 4, while our classification datasets are specified in Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Curves are shown with their standard variation for the randomized experiments which induced the most variation (Fig. 5 left and right).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Our experimental setup is specified at the beginning of Section 4

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our work conforms to the above code.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A].

    Justification: Our work has no direct societal impact.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our work poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Our dataset specification (Appendix E) cites the codes used to generate datasets [34], or directly points to the dataset repositories we used [28] and license terms are respected.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Our algorithm is described in Section 3.3. Our code will be made available on Github with documentation in case of acceptance.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: Our work does not involve crowdsourcing.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Our work does not involve study participants.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A].

    Justification: The development of our methodology does not involve LLMs.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.