LEARNING FROM FILTRATION SKETCHES

Anonymous authors
Paper under double-blind review

ABSTRACT

Methods from computational topology are widely used in machine learning to extract topological and geometric information from data. Their use, however, often involves several computational stages before a representation suitable for learning is obtained. We investigate whether the underlying mathematical object, a filtration of a simplicial complex built on the data, can itself serve as the basis of a learning representation. We introduce *filtration sketches*, which combine estimated spectral summaries at selected filtration levels with approximate persistent ranks between them. Both components are computed through sparse boundary-operator applications and randomized trace estimation, without explicit eigendecomposition or persistence-barcode computation. The resulting fixed-width sequence is processed by a sequence model for downstream learning. On molecular property prediction and point-cloud benchmarks, filtration sketches outperform topology-based baselines and match or outperform point-cloud-specific neural networks.

1 INTRODUCTION

Topological data analysis (TDA) (Carlsson, 2009) uses tools from computational topology to study the structure of data, with persistent homology (PH) (Edelsbrunner et al., 2002; Zomorodian & Carlsson, 2005) being arguably the most prominent method used by practitioners. In short, PH tracks the birth and death of homology classes along a filtration; for distance- or geometry-based filtrations, the resulting persistence barcode links topology to geometric scale. For learning, barcodes are typically converted into ML-friendly representations, such as persistence images, learned diagram layers, or data-adaptive vectorizations (Adams et al., 2017; Carrière et al., 2020; Royer et al., 2021). A substantial line of research therefore focuses on how to construct a principled interface between persistence summaries and learning methods.

Despite this *standard recipe*'s success, it remains unclear whether downstream learning requires exact persistence barcodes, or whether a compact summary of the filtration could provide a more useful representation. Passing to homology discards incidence information, and subsequent vectorization shapes what topological and geometric information remains accessible to the learning model. Hodge spectra capture structural information beyond Betti numbers (Horak & Jost, 2013); persistent spectral representations and filtration-indexed summaries have demonstrated the value of alternatives to barcode-based learning (Davies et al., 2023; Tola et al., 2025; Uddin et al., 2026). This motivates our question: *Can we combine local spectral information with approximate summaries of topological survival across filtration stages, without computing exact persistence barcodes?*

Contribution In this work, we explore *filtration sketches* (`FiltSketch`): ordered sequences that combine efficiently computable local Hodge spectral summaries with approximate persistent ranks between a small set of selected levels of a filtration. The Hodge stream captures kernel and positive spectral contributions, while the persistent-rank stream describes cross-sampling-point survival. Both streams use sparse boundary-matrix products and randomized trace estimation, avoiding explicit eigendecomposition and exact PH computation. A gated sequence model then consumes the two streams for downstream learning. Experiments on various benchmark datasets show that `FiltSketch` outperforms carefully tuned PH baselines, is compatible with different filtration techniques and, when selecting a suitable filtration, can even match neural network models on point cloud classification where benchmarks often do not even contain much discriminative topological information.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

2 Related work

Learning can enter topological representations at several stages. Given persistence barcodes, task-specific representations adapt their encoding to a prediction problem (Hofer et al., 2019; Carrière et al., 2020), while data-adaptive vectorizations such as ATOL construct an encoding from a collection of barcodes (Royer et al., 2021). Alternatively, Hofer et al. (2020) learn the filtration itself by differentiating through the PH computation. These approaches improve how persistence is extracted or represented, while retaining its computation as an intermediate step.

Another line of work replaces PH computation at inference with a learned surrogate. PI-Net (Som et al., 2020) predicts persistence images directly from signals or images, while PDGNN predicts extended persistence diagrams of graphs (Yan et al., 2022). Both use persistence-derived targets to train their predictors: the computational cost is incurred when constructing supervision and amortized over subsequent predictions.

Aside from exact and approximate PH computation, persistent Laplacians associate operators with inclusions between complexes. Their nullities recover persistent Betti numbers, while their positive spectra provide information beyond the ranks of the inclusion-induced homology maps (Mémoli et al., 2022). The molecular-property prediction experiments of Davies et al. (2023) demonstrate the predictive value of these spectral features. Yet, constructing persistent operators and extracting spectral information become costly as chain spaces and the number of queried filtration pairs grow.

More recently, some works have also explored learning representations directly from filtration stages or slices. TopER (Tola et al., 2025) summarizes the relationship between vertex and edge counts across graph-filtration stages through a linear fit. TopoFormer (Uddin et al., 2026) processes sequences of Betti numbers and simplex counts obtained from overlapping graph slices. While these representations retain information about the selected stages or slices, they do not explicitly encode the inclusion-induced homology maps of the original nested filtration. In particular, endpoint Betti numbers do not determine how many classes survive between the endpoints.

3 Preliminaries

Simplicial complexes, chains and homology Let V be a finite set. A simplicial complex $\mathcal{K}$ with vertices V is a collection of subsets of V closed under taking subsets. That is, if $\sigma \in \mathcal{K}$ and $\tau \subseteq \sigma$, then also $\tau \in \mathcal{K}$. Such subsets σ are called simplices, or q-simplices if $|\sigma| = q + 1$, and if $\tau \subseteq \sigma$ we call τ a face of σ and σ a coface of τ.

We assign orientations to the simplices of $\mathcal{K}$ by enumerating its vertices $V = \{v_1, \dots, v_N\}$, thereby each q-simplex $\sigma = \{v_{i_0}, \dots, v_{i_q}\}$ with $i_0 < \cdots < i_q$ gets a canonical representative $[v_{i_0}, \dots, v_{i_q}]$. The real q-chain space $C_q(\mathcal{K}; \mathbb{R})$ is then the abstract vector space over $\mathbb{R}$ with these representatives as basis. The structure of the simplicial complex is encoded in the boundary operators $\partial_q : C_q(\mathcal{K}; \mathbb{R}) \to C_{q-1}(\mathcal{K}; \mathbb{R})$ which for $q \geq 1$ are defined by linear extension of their action on basis vectors

$$\partial_q [v_{i_0}, \dots, v_{i_q}] = \sum_{j=0}^{q} (-1)^j [v_{i_0}, \dots, \widehat{v}_{i_j}, \dots, v_{i_q}] \ , \tag{1}$$

with $\widehat{v}_{i_j}$ denoting the deletion of the vertex v_{i_j}, and for $q = 0$ we set $\partial_0 = 0$. The elements of $Z_q(\mathcal{K}; \mathbb{R}) := \ker \partial_q \subseteq C_q(\mathcal{K}; \mathbb{R})$ are called q-cycles, and the elements of $B_q(\mathcal{K}; \mathbb{R}) := \operatorname{im} \partial_{q+1} \subseteq C_q(\mathcal{K}; \mathbb{R})$ are called q-boundaries. The composition of two consecutive boundary maps vanishes, i.e., $\operatorname{im} \partial_{q+1} \subseteq \ker \partial_q$, so every boundary is a cycle. Identifying cycles that differ only by a boundary gives the q-th homology group $H_q(\mathcal{K}; \mathbb{R}) := Z_q(\mathcal{K}; \mathbb{R}) / B_q(\mathcal{K}; \mathbb{R})$ and the q-th Betti number $\beta_q(\mathcal{K}) := \dim H_q(\mathcal{K}; \mathbb{R})$. Homology is functorial, meaning that a map f between simplicial complexes induces a map f_* between their homology groups.

Hodge decomposition Equipping the chain spaces $C_q(\mathcal{K}; \mathbb{R})$ with an inner product yields an adjoint boundary operator $\partial_q^* : C_{q-1}(\mathcal{K}; \mathbb{R}) \to C_q(\mathcal{K}; \mathbb{R})$ implicitly defined by $\langle \partial_q^* x, y \rangle_q = \langle x, \partial_q y \rangle_{q-1}$. The *Hodge Laplacian* $\Delta_q : C_q(\mathcal{K}; \mathbb{R}) \to C_q(\mathcal{K}; \mathbb{R})$ is then defined as $\Delta_q = \partial_{q+1} \circ \partial_{q+1}^* + \partial_q^* \circ \partial_q$, and the combinatorial Hodge theorem gives the orthogonal decomposition of the chain space

$$C_q(\mathcal{K}; \mathbb{R}) = \operatorname{im} \partial_{q+1} \oplus \operatorname{im} \partial_q^* \oplus \ker \Delta_q \ . \tag{2}$$

By the definition of the adjoint boundary operator, cycles are orthogonal to $\operatorname{im} \partial_q^*$, so the decomposition uniquely expresses every cycle as a sum of a boundary and a harmonic chain in $\ker \Delta_q$. Consequently, every homology class has a unique harmonic representative giving an isomorphism $H_q(\mathcal{K}; \mathbb{R}) \simeq \ker \Delta_q$. In addition to the topological information encoded in its kernel, the Hodge Laplacian encodes further geometric properties of the complex $\mathcal{K}$. For example, at $q = 0$, Δ_0 is the graph Laplacian of the 1-skeleton, and its smallest positive eigenvalue on each connected component quantifies how well-connected that component is (Fiedler, 1973).

Matrix representation Since chain spaces $C_q(\mathcal{K}; \mathbb{R})$ are real vector spaces of dimension $n_q = |\{\sigma \in \mathcal{K} \colon \sigma \text{ is } q\text{-simplex}\}|$, we can identify them with $\mathbb{R}^{n_q}$ by selecting an order on the q-simplices, and thus on the canonical basis from before. Under this identification, the boundary map $\partial_q : C_q(\mathcal{K}; \mathbb{R}) \to C_{q-1}(\mathcal{K}; \mathbb{R})$ becomes a matrix $\mathbf{B}_q \in \{-1, 0, 1\}^{n_{q-1} \times n_q}$. Similarly, after equipping the chain spaces with the inner product that makes this basis orthonormal, the adjoint boundary operator $\partial_q^* : C_{q-1}(\mathcal{K}; \mathbb{R}) \to C_q(\mathcal{K}; \mathbb{R})$ is represented by the matrix transpose $\mathbf{B}_q^\top$, and thus the Hodge Laplacian and the Hodge decomposition are represented by

$$\mathbf{L}_q = \mathbf{B}_q^\top \mathbf{B}_q + \mathbf{B}_{q+1} \mathbf{B}_{q+1}^\top \qquad \text{and} \qquad \mathbb{R}^{n_q} = \operatorname{im} \mathbf{B}_{q+1} \oplus \operatorname{im} \mathbf{B}_q^\top \oplus \ker \mathbf{L}_q \ . \tag{3}$$

Filtrations and persistent homology Given a finite simplicial complex $\mathcal{K}$ and a function $f : \mathcal{K} \to \mathbb{R}$ that satisfies $f(\tau) \leq f(\sigma)$ whenever $\tau \subseteq \sigma \in \mathcal{K}$, its sublevel sets $\mathcal{K}_f(t) := \{\sigma \in \mathcal{K} \colon f(\sigma) \leq t\} \subseteq \mathcal{K}$ are again simplicial complexes, and moreover $\mathcal{K}_f(t) \subseteq \mathcal{K}_f(s)$ whenever $t \leq s$. As f can attain at most as many distinct values $\xi_0 < \cdots < \xi_M$ as there are simplices in $\mathcal{K}$, this gives a finite sequence

$$\mathcal{K}^0 \subseteq \mathcal{K}^1 \subseteq \cdots \subseteq \mathcal{K}^M = \mathcal{K} \ , \text{ with } \mathcal{K}^j = \mathcal{K}_f(\xi_j) \tag{4}$$

of simplicial complexes, called a filtration of $\mathcal{K}$, and, accordingly, f is called a filtration function. For example, given a finite metric space (X, d), the sublevel sets of the function $f : \sigma \mapsto \max_{x,y \in \sigma} d(x, y)$ form the Vietoris–Rips filtration of the full simplex $\{\sigma \colon \sigma \subseteq X\}$ on X.

The inclusions of subcomplexes in Eq. (4) induce linear maps in homology, and we can track homology classes through the sequence

$$H_q(\mathcal{K}^0; \mathbb{R}) \longrightarrow H_q(\mathcal{K}^1; \mathbb{R}) \longrightarrow \cdots \longrightarrow H_q(\mathcal{K}^{M-1}; \mathbb{R}) \longrightarrow H_q(\mathcal{K}^M; \mathbb{R}) \ . \tag{5}$$

Specifically, the images of the maps $H_q(\mathcal{K}^i; \mathbb{R}) \to H_q(\mathcal{K}^j; \mathbb{R})$ are called the persistent homology groups $H_q^{i,j}(\mathcal{K}; \mathbb{R})$ and their ranks

$$\beta_q^{i,j}(\mathcal{K}) = \operatorname{rank}(H_q(\mathcal{K}^i; \mathbb{R}) \longrightarrow H_q(\mathcal{K}^j; \mathbb{R})) = \dim(H_q^{i,j}(\mathcal{K}; \mathbb{R})) \tag{6}$$

are called *persistent ranks* or persistent Betti numbers. They measure how many independent homology classes from $H_q(\mathcal{K}^i; \mathbb{R})$ remain independent and nontrivial in $H_q(\mathcal{K}^j; \mathbb{R})$; information that is not determined by the two endpoint Betti numbers. The collection of all persistent ranks carries the same information as the persistence barcode of the filtration which is the quantity that is typically used in vectorized form as a representation for learning approaches.

4 Filtration Sketches

Our goal is to transform a finite filtration into an ordered, fixed-width representation for learning. At each of T sampled filtration values, we construct two complementary descriptor streams: a **Hodge stream**, which summarizes spectral structure within a single complex, and a **persistent-rank stream**, which summarizes survival of homology classes between pairs of complexes along the filtration. We first define both descriptors for a single complex and for a pair of nested complexes (Sections 4.1 and 4.2), and then describe how we sample the filtration and assemble the descriptors into a sequence (Section 4.3).

4.1 Local Hodge spectral summaries

We want to use the spectrum of Hodge Laplacians as features for learning. However, as the number of simplices in a complex, and thus the dimensions of the Laplacian can be very large, we want to avoid computing the full spectrum by diagonalization. Instead, we aim to represent its spectral

density using a finite number of spectral sums (Han et al., 2017) as feature vectors. The latter are quantities of the form $\operatorname{tr} f(\mathbf{L}) = \sum_i f(\lambda_i)$, where f is a polynomial and λ_i denote the eigenvalues of $\mathbf{L}$. They can be efficiently estimated by randomized trace estimation (Han et al., 2017) using only sparse matrix-vector products.

To focus on $\ker \mathbf{L}_q \simeq H_q(\mathcal{K}; \mathbb{R})$ and the low end of the spectrum, we use transformations of the form $f_k(\mathbf{L}_q) = (\mathbf{I} - \alpha \mathbf{L}_q)^k$ with $\alpha \in (0, 1]$. After normalizing $\mathbf{L}_q$ to $\overline{\mathbf{L}}_q$ such that its spectrum is a subset of $[0, 1]$, $f_k(\overline{\mathbf{L}}_q)$ converges exponentially in k to the orthogonal projection onto $\ker \overline{\mathbf{L}}_q = \ker \mathbf{L}_q$; f_k preserves eigenvectors and the zero eigenvalues are mapped to one while the others are mapped to $(1 - \alpha\lambda_i)^k$ which decays exponentially in k. Since $(\mathbf{I} - \alpha \overline{\mathbf{L}}_q)^k \mathbf{x}$ performs k Euler steps with step size α of the heat equation $\dot{\mathbf{x}} = -\overline{\mathbf{L}}_q \mathbf{x}$, we call the matrices $f_k(\overline{\mathbf{L}}_q)$ diffusion filters and their traces *diffusion moments*.

Normalizing Hodge Laplacians We normalize the Hodge Laplacian such that it has spectrum in $[0, 1]$ while sharing an eigenbasis with the original operator and preserving its harmonic subspace.

To this end, we first normalize the boundary matrices. Let $q \geq 1$ and assume that the simplicial complex $\mathcal{K}$ has a q-simplex. Recall that the (i, j)-th entry of the boundary matrix $\mathbf{B}_q$ is non-zero if and only if the i-th $(q-1)$-simplex is a face of the j-th q-simplex. Since every q-simplex has exactly $(q+1)$ faces of dimension $q-1$, this implies that all columns of $\mathbf{B}_q$ have exactly $q+1$ non-zero entries, while the number of non-zero entries of a row is given by the number of cofaces of dimension q that a $(q-1)$-simplex has. Moreover, since the entries of $\mathbf{B}_q$ are in $\{-1, 0, 1\}$, we have $\|\mathbf{B}_q\|_1 = q + 1$ and $\|\mathbf{B}_q\|_\infty = d_q^{\max}$ is the maximal number of q-cofaces of a $(q-1)$-simplex. Thus,

$$\|\mathbf{B}_q\|_2^2 \leq \|\mathbf{B}_q\|_1 \|\mathbf{B}_q\|_\infty = (q+1) d_q^{\max} \quad . \tag{7}$$

This allows introducing the *normalized boundary matrix* $\overline{\mathbf{B}}_q := {}^1\!/\!\sqrt{(q+1) d_q^{\max}}\, \mathbf{B}_q$ which has spectral norm $\|\overline{\mathbf{B}}_q\|_2 \leq 1$. At the lowest and highest dimensions of the chain complex the missing boundary maps are taken to be zero: $\overline{\mathbf{B}}_0 = \mathbf{B}_0 = \mathbf{0}$ and $\overline{\mathbf{B}}_{d+1} = \mathbf{B}_{d+1} = \mathbf{0}$, where $d = \dim \mathcal{K}$. Normalizing the boundary matrices preserves their kernels and images, as well as $\overline{\mathbf{B}}_q \overline{\mathbf{B}}_{q+1} = \mathbf{0}$. It therefore leaves the subspaces in the Hodge decomposition unchanged.

We define the normalized lower, upper, and full Hodge Laplacians as

$$\overline{\mathbf{L}}_q^{\text{down}} = \overline{\mathbf{B}}_q^\top \overline{\mathbf{B}}_q, \qquad \overline{\mathbf{L}}_q^{\text{up}} = \overline{\mathbf{B}}_{q+1} \overline{\mathbf{B}}_{q+1}^\top, \qquad \overline{\mathbf{L}}_q^{\text{full}} = \overline{\mathbf{L}}_q^{\text{down}} + \overline{\mathbf{L}}_q^{\text{up}} \quad . \tag{8}$$

For a chain with coordinates $\mathbf{x}$, the lower energy $\mathbf{x}^\top \overline{\mathbf{L}}_q^{\text{down}} \mathbf{x} = \|\overline{\mathbf{B}}_q \mathbf{x}\|_2^2$ vanishes exactly on cycles and the upper energy $\mathbf{x}^\top \overline{\mathbf{L}}_q^{\text{up}} \mathbf{x} = \|\overline{\mathbf{B}}_{q+1}^\top \mathbf{x}\|_2^2$ exactly on chains orthogonal to boundaries. Hence,

$$\ker(\overline{\mathbf{L}}_q^{\text{full}}) = \ker(\overline{\mathbf{B}}_q) \cap \ker(\overline{\mathbf{B}}_{q+1}^\top) \cong H_q(\mathcal{K}; \mathbb{R}) \quad . \tag{9}$$

Moreover, since boundaries are cycles, the lower energy vanishes on $\operatorname{im} \overline{\mathbf{B}}_{q+1}$ and the upper energy on $\operatorname{im} \overline{\mathbf{B}}_q^\top$. Hence, each operator acts only on its own summand of the Hodge decomposition Eq. (3), and thus $\|\overline{\mathbf{L}}_q\|_2 = \max(\|\overline{\mathbf{B}}_q\|_2^2, \|\overline{\mathbf{B}}_{q+1}\|_2^2) \leq 1$.

Diffusion moments We apply the diffusion filters to the lower, upper, and full Hodge Laplacians of Eq. (8) giving the normalized diffusion moments

$$m_{q,k}^\star(\mathcal{K}) := \frac{1}{n_q(\mathcal{K})} \operatorname{tr}\left((\mathbf{I} - \alpha_\star \overline{\mathbf{L}}_q^\star)^k \right) \quad , \qquad \alpha_{\text{down}} = \alpha_{\text{up}} = 1,\ \alpha_{\text{full}} = \tfrac{1}{2} \quad , \tag{10}$$

where $\star \in \{\text{down}, \text{up}, \text{full}\}$ and $k \geq 1$. The factor $\alpha_{\text{full}} = 1/2$ for the full Laplacian is a design choice to slow attenuation of positive spectral contributions. A fixed selection of powers gives a fixed number of descriptors, independently of n_q, where larger powers increasingly emphasize the zero and smallest positive eigenvalues. For an empty chain space, we set $m_{q,k}^\star(\mathcal{K}) = 0$. For the following discussion we abbreviate $m_{q,k}^\star = m_{q,k}^\star(\mathcal{K})$.

Since $f_k(\overline{\mathbf{L}}_q^\star)$ converges to the orthogonal projector onto $\ker \overline{\mathbf{L}}_q^\star$, taking traces yields, as $k \to \infty$,

$$\begin{aligned} n_q m_{q,k}^{\text{down}} &\to \operatorname{nullity} \mathbf{B}_q = \dim Z_q, & n_q\big(1 - m_{q,k}^{\text{down}}\big) &\to \operatorname{rank} \mathbf{B}_q = \dim B_{q-1}, \\ n_q m_{q,k}^{\text{up}} &\to \operatorname{nullity} \mathbf{B}_{q+1}^\top = \dim B_q^\perp, & n_q m_{q,k}^{\text{full}} &\to \operatorname{nullity} \overline{\mathbf{L}}_q^{\text{full}} = \beta_q \quad . \end{aligned} \tag{11}$$

These limits identify the kernel contributions to the moments: cycle directions for the lower operator, directions orthogonal to boundaries for the upper operator, and harmonic directions representing homology for the full operator. More precisely, each zero eigenvalue contributes 1 to $\operatorname{tr} f_k(\overline{\mathbf{L}}_q^\star)$, while each positive eigenvalue $\lambda \geq \gamma$ contributes $(1-\alpha_\star\lambda)^k \in [0, (1-\alpha_\star\gamma)^k]$, where γ is the smallest positive eigenvalue of $\overline{\mathbf{L}}_q^\star$. Hence,

$$0 \leq n_q m_{q,k}^\star - \dim\ker \overline{\mathbf{L}}_q^\star \leq \operatorname{rank}(\overline{\mathbf{L}}_q^\star)\,(1-\alpha_\star\gamma)^k \quad , \tag{12}$$

so the convergence is exponential in k with a rate $1-\alpha_\star\gamma$.

At *finite depth*, each moment combines the corresponding kernel dimension, normalized by n_q, with contributions from positive Hodge eigenvalues. We retain several finite powers to capture this spectral information alongside the kernel contributions, rather than only the limiting dimensions. Thus, complexes with identical Betti numbers can have different descriptors. However, a finite selection of powers only summarizes the spectral density rather than determining it, and it recovers kernel dimensions, including Betti numbers, only up to the error in Eq. (12).

Hodge stream We estimate the diffusion moments in Eq. (10) using the Hutchinson trace estimator (Hutchinson, 1989; Avron & Toledo, 2011) with R independent Rademacher probes $\mathbf{z}_r$. That is, we set

$$\widehat{m}_{q,k}^\star(\mathcal{K}) := \frac{1}{R\,n_q(\mathcal{K})} \sum_{r=1}^{R} \mathbf{z}_r^\top (\mathbf{I} - \alpha_\star \overline{\mathbf{L}}_q^\star)^k \mathbf{z}_r, \quad \text{if } n_q(\mathcal{K}) > 0 \ , \tag{13}$$

and $\widehat{m}_{q,k}^\star(\mathcal{K}) = 0$ otherwise. This estimator is unbiased and can be efficiently computed from sparse matrix-vector products. It gives one scalar feature per dimension q, diffusion depth k, and normalized Laplacian $\overline{\mathbf{L}}_q^\star \in \{\overline{\mathbf{L}}_q^{\text{down}}, \overline{\mathbf{L}}_q^{\text{up}}, \overline{\mathbf{L}}_q^{\text{full}}\}$. Denoting by $\mathcal{Q}$ and $\mathcal{P}$ the sets of selected dimensions q and depths k, respectively, this gives a $3|\mathcal{Q}||\mathcal{P}|$-dimensional descriptor for the structure of a complex $\mathcal{K}$, which we denote by

$$\mathbf{h}(\mathcal{K}) := \big(\widehat{m}_{q,k}^\star(\mathcal{K}) \colon q \in \mathcal{Q}, k \in \mathcal{P}, \star \in \{\text{down}, \text{up}, \text{full}\}\big) \in \mathbb{R}^{3|\mathcal{Q}||\mathcal{P}|} \ . \tag{14}$$

4.2 Approximate persistent ranks

The Hodge features from Section 4.1 describe individual complexes but do not encode relations between different complexes, e.g., those that come from a filtration. For example, there may be situations when the Betti numbers $\beta_1(\mathcal{K}) = \beta_1(\mathcal{K}') = 1$ of two complexes $\mathcal{K} \subseteq \mathcal{K}'$ agree, and thus that both complexes contain exactly one independent loop. However, these loops might be of different origin, as the void bounded by the loop in $\mathcal{K}$ might be filled in $\mathcal{K}'$. More formally, a non-zero homology class in $H_1(\mathcal{K};\mathbb{R})$ may be mapped to zero in $H_1(\mathcal{K}';\mathbb{R})$ under the map $H_1(\mathcal{K};\mathbb{R}) \to H_1(\mathcal{K}';\mathbb{R})$ induced by the inclusion $\mathcal{K} \hookrightarrow \mathcal{K}'$. Hence, to resolve such effects, we need to study the ranks of the induced map, i.e., the persistent ranks or persistent Betti numbers

$$\beta_q(\mathcal{K} \hookrightarrow \mathcal{K}') := \operatorname{rank}(H_q(\mathcal{K};\mathbb{R}) \longrightarrow H_q(\mathcal{K}';\mathbb{R})) \quad . \tag{15}$$

Exact persistent ranks The persistent rank can be computed from the source boundary matrix $\mathbf{B}_q(\mathcal{K})$ and target boundary matrix $\mathbf{B}_{q+1}(\mathcal{K}')$ via the formula (see Appendix C.2 for a derivation)

$$\beta_q(\mathcal{K} \hookrightarrow \mathcal{K}') = \underbrace{\dim\ker(\mathbf{B}_q(\mathcal{K}))}_{\text{source cycles}} - \underbrace{\dim(\operatorname{im}(\mathbf{B}_{q+1}(\mathcal{K}')) \cap C_q(\mathcal{K};\mathbb{R}))}_{\text{target boundaries supported on the source}} . \tag{16}$$

To evaluate the intersection term, we order the simplices of $\mathcal{K}'$ so that those already present in $\mathcal{K}$ come first. In the resulting bases of $C_{q+1}(\mathcal{K}';\mathbb{R})$ and $C_q(\mathcal{K}';\mathbb{R})$ the boundary matrix has the form

$$\mathbf{B}_{q+1}(\mathcal{K}') = \begin{bmatrix} \mathbf{B}_{q+1}(\mathcal{K}) & \mathbf{X}_{q+1}(\mathcal{K} \hookrightarrow \mathcal{K}') \\ \mathbf{0} & \mathbf{Y}_{q+1}(\mathcal{K} \hookrightarrow \mathcal{K}') \end{bmatrix} . \tag{17}$$

We drop the dependency of $\mathbf{Y}_{q+1}$ and derived quantities on the inclusion from now on, unless we want to stress it in key equations. The bottom left block is zero since every face of a source simplex must be in the source complex $\mathcal{K}$. The columns of $\begin{bmatrix} \mathbf{0} \\ \mathbf{Y}_{q+1} \end{bmatrix}$ span the image of the projection map from

$\operatorname{im} \mathbf{B}_{q+1}(\mathcal{K}')$ onto the subspace spanned by the q-simplices of $\mathcal{K}' \setminus \mathcal{K}$. The rank of this projection therefore is exactly $\operatorname{rank} \mathbf{Y}_{q+1}$. The kernel of this map is just $\operatorname{im} \mathbf{B}_{q+1}(\mathcal{K}') \cap C_q(\mathcal{K}; \mathbb{R})$. Rank-nullity then implies

$$\dim \operatorname{im} \mathbf{B}_{q+1}(\mathcal{K}') = \operatorname{rank} \mathbf{Y}_{q+1} + \dim \left(\operatorname{im} \mathbf{B}_{q+1}(\mathcal{K}') \cap C_q(\mathcal{K}; \mathbb{R}) \right) \tag{18}$$

and substituting into Eq. (16) gives

$$\beta_q(\mathcal{K} \hookrightarrow \mathcal{K}') = \operatorname{nullity} \mathbf{B}_q(\mathcal{K}) - \operatorname{rank} \mathbf{B}_{q+1}(\mathcal{K}') + \operatorname{rank} \mathbf{Y}_{q+1}(\mathcal{K} \hookrightarrow \mathcal{K}') \ . \tag{19}$$

Finite-depth approximation The first two terms in Eq. (19) depend only on the individual complexes. The first is the dimension of the q-cycle space $\operatorname{nullity} \mathbf{B}_q(\mathcal{K}) = \dim Z_q(\mathcal{K}; \mathbb{R})$, and the second is the dimension of the q-boundary space $\operatorname{rank} \mathbf{B}_{q+1}(\mathcal{K}') = \dim B_q(\mathcal{K}'; \mathbb{R})$. Both can be approximated via the previously introduced normalized lower-Hodge moments since, by Eq. (11),

$$n_q(\mathcal{K})\, m^{\text{down}}_{q,k}(\mathcal{K}) \longrightarrow \operatorname{nullity} \mathbf{B}_q(\mathcal{K}) \ , \quad n_{q+1}(\mathcal{K}')\big(1 - m^{\text{down}}_{q+1,k}(\mathcal{K}')\big) \longrightarrow \operatorname{rank} \mathbf{B}_{q+1}(\mathcal{K}') \tag{20}$$

as $k \to \infty$. The quantity $\operatorname{rank} \mathbf{Y}_{q+1}$ depends on how $\mathcal{K}$ sits inside $\mathcal{K}'$ and requires a pair-specific operator. We normalize $\mathbf{Y}_{q+1}$ by the same incidence-count bound used for full boundary matrices (cf. Section 4.1):

$$\overline{\mathbf{Y}}_{q+1} := \frac{1}{\sqrt{\max\{1, (q+2)\|\mathbf{Y}_{q+1}\|_\infty\}}} \mathbf{Y}_{q+1} \ , \tag{21}$$

where $\|\mathbf{Y}_{q+1}\|_\infty$ denotes the maximum absolute row sum, defined as zero when $\mathcal{K}'$ has no new q-simplices, in which case $\mathbf{Y}_{q+1}$ has no rows. Each column of $\mathbf{Y}_{q+1}$ has at most $q+2$ nonzero entries, so $\|\overline{\mathbf{Y}}_{q+1}\|_2 \le 1$.

We denote by $\mathbf{G}_{q+1}$ either Gram matrix $\overline{\mathbf{Y}}_{q+1}^\top \overline{\mathbf{Y}}_{q+1}$ or $\overline{\mathbf{Y}}_{q+1} \overline{\mathbf{Y}}_{q+1}^\top$ and its dimension by N_{q+1}. In analogy with Eq. (10), let

$$a_{q+1,k} := N_{q+1} - \operatorname{tr}\big[(\mathbf{I} - \mathbf{G}_{q+1})^k\big] \, . \tag{22}$$

Each positive Gram eigenvalue λ contributes $1 - (1-\lambda)^k$ which approaches one, while zeros do not contribute. Thus, either Gram choice yields the same exact finite-depth surrogate, and

$$a_{q+1,k} \to \operatorname{rank} \mathbf{Y}_{q+1} \qquad \text{as } k \to \infty \ . \tag{23}$$

For a zero-dimensional Gram matrix, we set $a_{q+1,k} = 0$. Combining the three terms gives

$$\beta_{q,k}(\mathcal{K} \hookrightarrow \mathcal{K}') := n_q(\mathcal{K})\, m^{\text{down}}_{q,k}(\mathcal{K}) - n_{q+1}(\mathcal{K}')\big(1 - m^{\text{down}}_{q+1,k}(\mathcal{K}')\big) + a_{q+1,k}(\mathcal{K} \hookrightarrow \mathcal{K}') \ . \tag{24}$$

At finite depth, this is a spectral surrogate for the persistent rank; it need not be an integer or satisfy the bounds of an exact rank.

Persistent-rank stream We estimate the finite-depth quantities above using the same randomized trace estimator with R independent Rademacher probes $\mathbf{z}_r$ as in Section 4.1. Specifically, we use

$$\widehat{a}_{q+1,k}(\mathcal{K} \hookrightarrow \mathcal{K}') := N_{q+1} - \frac{1}{R} \sum_{r=1}^{R} \mathbf{z}_r^\top (\mathbf{I} - \mathbf{G}_{q+1})^k \mathbf{z}_r \ , \tag{25}$$

with $\widehat{a}_{q+1,k}(\mathcal{K} \hookrightarrow \mathcal{K}') = 0$ when $N_{q+1} = 0$. The unbiased estimate $\widehat{\beta}_{q,k}(\mathcal{K} \hookrightarrow \mathcal{K}')$ is then obtained from Eq. (24) by replacing each trace by its estimate. Again, the probe count (for trace estimation) controls stochastic estimation error, while the diffusion depth controls the underlying spectral approximation.

Denoting by $\mathcal{Q}$ and $\mathcal{P}$ the sets of selected dimensions q and depths k, as for the Hodge features, this gives a $|\mathcal{Q}||\mathcal{P}|$-dimensional feature describing how $\mathcal{K}$ sits inside $\mathcal{K}'$, which we denote by

$$\mathbf{p}(\mathcal{K} \hookrightarrow \mathcal{K}') := \Big(\psi\big(\widehat{\beta}_{q,k}(\mathcal{K} \hookrightarrow \mathcal{K}')\big) \colon q \in \mathcal{Q}, k \in \mathcal{P} \Big) \in \mathbb{R}^{|\mathcal{Q}||\mathcal{P}|} \ , \tag{26}$$

where ψ denotes the clipping and log-transform defined in Appendix B.

4.3 TWO-STREAM FILTRATION SKETCH

We represent a filtration as an ordered sequence combining the Hodge and persistent-rank streams. To obtain this sequence, we sample the filtration at T filtration levels.

For learning, these filtration levels are determined by the training data and shared with the validation and test data. Denoting by a and b the smallest and largest filtration values occurring across all training filtrations, we choose equally spaced filtration levels $t_s = a + {}^{s}\!/_{T}(b-a)$ with $s = 1, \dots, T$. For each filtration, we retain the subfiltration determined by these levels $\mathcal{K}_f(t_1) \subseteq \mathcal{K}_f(t_2) \subseteq \cdots \subseteq \mathcal{K}_f(t_T)$.

At each filtration level t_s, we record three kinds of features: (1) the Hodge features $\mathbf{h}_s = \mathbf{h}(\mathcal{K}_f(t_s))$ summarize the topology and geometry of the complex at that filtration level; (2) the persistent-rank features $\mathbf{p}(\mathcal{K}_f(t_s) \hookrightarrow \mathcal{K}_f(t_{s+\ell}))$ summarize the survival of homology classes from level s to level $s+\ell$. Here ℓ ranges over a fixed set $\mathcal{L}$ of positive integer lags. We concatenate these pair descriptors over $\ell \in \mathcal{L}$ to obtain $\mathbf{p}_s$, padding unavailable pairs with $s + \ell > T$ with zeros. Finally, (3) the filtration coordinates $\mathbf{c}_s \in \mathbb{R}^2$ record the fraction of distinct filtration values already reached at t_s as their first coordinate, and the threshold t_s itself as their second. These coordinates distinguish progression through the filtration from its physical scale.

The complete per-sampling-point descriptor and final filtration sketch representation, denoted as `FiltSketch`, are (with $F = 3|\mathcal{Q}||\mathcal{P}| + |\mathcal{Q}||\mathcal{L}||\mathcal{P}| + 2$)

$$\mathbf{x}_s := \mathrm{concat}(\mathbf{h}_s, \mathbf{p}_s, \mathbf{c}_s) \ \text{, and } \ \mathbf{X} := [\mathbf{x}_1, \dots, \mathbf{x}_T] \in \mathbb{R}^{F \times T} \ . \tag{27}$$

4.4 DOWNSTREAM SEQUENCE MODELING

We process the filtration sketch $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_T]$ with a temporal convolutional network (TCN), where $\mathbf{x}_s = \mathrm{concat}(\mathbf{h}_s, \mathbf{p}_s, \mathbf{c}_s)$. The convolutional filters capture patterns along the filtration axis, while dilation combines local changes with information from more widely separated filtration levels. We use noncausal convolutions because the complete sketch is available at prediction time. For fixed width and depth, their computational cost grows linearly with the number of filtration levels.

Gating As different data might require focusing on one stream more than the other, we implement a simple gating mechanism. First, we map the streams in $\mathbf{x}_s$ separately before combining them. It uses both feature values and first differences: e.g., $\delta\mathbf{p}_s = \mathbf{p}_s - \mathbf{p}_{s-1}$ for $s > 1$, with $\delta\mathbf{p}_1 = \mathbf{0}$. Values describe the state at each filtration level, while differences expose changes between successive levels. Writing $\mathbf{h}^\star_s$, $\star \in \{\text{down}, \text{up}, \text{full}\}$, for the three Hodge groups within $\mathbf{x}_s$, the adapter computes

$$\mathbf{u}^{(0)}_s = \mathrm{LN}\Big(\mathbf{W}_p\mathbf{p}_s + \mathbf{V}_p\delta\mathbf{p}_s + \phi(\mathbf{c}_s) + \sum_{\star \in \{\text{down},\text{up},\text{full}\}} \tanh(g_\star)\big(\mathbf{W}_\star\mathbf{h}^\star_s + \mathbf{V}_\star\delta\mathbf{h}^\star_s\big)\Big). \tag{28}$$

All projections map to a common latent width, ϕ is a two-layer coordinate embedding with a SiLU activation, and LN denotes layer normalization. Each $g_\star$ is a learned scalar shared across filtration levels and samples. Initializing these parameters to zero makes persistent ranks and filtration coordinates the initial information path, while allowing each Hodge group to contribute during training. The resulting latent sequence is then processed by the TCN.

Readout We read out the sequence after every second convolutional layer and after the final one. At each such layer, we summarize the layer-normalized states by concatenating attention pooling, mean pooling, coordinate-wise maximum pooling, and the final state. These capture learned filtration level importance, average responses, peak responses, and the state at the final sampled complex. Since layers at different depths have different receptive fields, we combine their summaries with learned coordinate-wise softmax weights over the layers. A two-layer MLP head with SiLU activation and dropout either yields logits or regression outputs.

5 EXPERIMENTS

We evaluate filtration sketches (`FiltSketch`) across different benchmark datasets and different filtration choices. Notably, throughout all experiments, the parametrization & training protocol remains the same; only on `QM7b` do we adjust the sampling resolution to the filtration length.

Table 1: Per-target test MAE on `QM7b`, reported as mean ± sample std. dev. across five splits. TopoFormer† denotes our inverse-Coulomb Topo-Scan (Uddin et al., 2026) adaptation, while Davies et al. (2023) report the persistent-Laplacian *Filtration-1* result. Relative improvements are computed as $100(\mathrm{MAE}_{\mathrm{baseline}} - \mathrm{MAE}_{\mathrm{ours}})/\,\mathrm{MAE}_{\mathrm{baseline}}$; positive values favor `FiltSketch`.

Target	`FiltSketch`	TopoFormer†	Davies et al.	Rel. MAE red. *vs* `TopoFormer` (%)	Rel. MAE red. *vs* Davies et al. (%)
T_1	20.11 ± 0.91	25.18 ± 4.70	**12.35 ± 0.60**	+20.2	-62.8
T_2	**1.44 ± 0.03**	1.71 ± 0.04	1.73 ± 0.04	+15.6	+16.6
T_3	**0.08 ± 0.00**	0.09 ± 0.00	0.10 ± 0.00	+13.0	+20.6
T_4	**0.30 ± 0.01**	0.42 ± 0.01	0.39 ± 0.01	+28.3	+22.8
T_5	**0.31 ± 0.01**	0.49 ± 0.03	0.36 ± 0.01	+36.8	+13.9
T_6	**0.40 ± 0.01**	0.65 ± 0.02	0.43 ± 0.01	+39.4	+7.1
T_7	**0.33 ± 0.01**	0.45 ± 0.01	0.41 ± 0.01	+27.8	+19.6
T_8	**0.33 ± 0.02**	0.55 ± 0.03	0.39 ± 0.01	+38.6	+14.6
T_9	**0.25 ± 0.00**	0.31 ± 0.00	0.27 ± 0.01	+19.6	+9.2
T_{10}	**0.22 ± 0.01**	0.28 ± 0.01	0.23 ± 0.01	+19.4	+1.5
T_{11}	**0.27 ± 0.00**	0.34 ± 0.00	0.31 ± 0.01	+20.6	+10.3
T_{12}	**0.19 ± 0.01**	0.23 ± 0.00	0.20 ± 0.00	+16.7	+4.0
T_{13}	**0.23 ± 0.01**	0.32 ± 0.01	0.35 ± 0.01	+28.0	+34.4
T_{14}	**0.18 ± 0.01**	0.23 ± 0.02	0.28 ± 0.01	+24.0	+37.7

Parametrization and training We use $Q \in \{0, 1, 2\}$ homological dimensions, $R = 8$ Rademacher probes for trace estimation, diffusion depths of 1,2,4 and 8 and five lags (see experiment descriptions) to compute the persistent-rank stream. The number of TCN layers is set to cover the sequence length T. We optimize either cross-entropy or the MSE for 150 epochs with a batch size of 32, using AdamW with weight decay of 1e-3 and a cosine learning rate scheduler, starting at 5e-4.

Molecular property prediction We follow the `QM7b` Coulomb-matrix Filtration-1 setup of Davies et al. (2023). The dataset comprises 7,211 molecules with 14 regression targets. All atoms enter the filtration at time zero, and an edge between atoms i and j enters at the inverse off-diagonal Coulomb value $1/C_{ij} = \|\mathbf{r}_i - \mathbf{r}_j\|/(Z_i Z_j)$ (with $\mathbf{r}_i \in \mathbb{R}^3$ denoting an atom's spatial coordinate and Z_i denoting its atomic number); overall, this yields a Vietoris–Rips filtration induced by the inverse-Coulomb dissimilarities. We compare `FiltSketch`, trained jointly on all targets, with the persistent-Laplacian Filtration-1 results and a TopoFormer-style baseline (Uddin et al., 2026), adapted to the same filtration. For the latter, we fit a shared quantile grid on the training molecules of each split, form overlapping interlevel graph slices, encode each slice by $(\beta_0, \beta_1, |V|, |E|)$, and process the resulting sequence with the released Transformer architecture. This adaptation excludes the additional molecular-fingerprint branch, thereby comparing the *filtration representations* directly. We report per-target test mean absolute error (MAE) as the mean and sample standard deviation across five splits in Table 1. `FiltSketch` (using $T = 128$, due to the relatively small molecules and thus short filtration lengths; lags 1,2,8,32,64) improves upon the adapted TopoFormer baseline on all 14 targets, reducing MAE by 13.0–39.4%. It also outperforms the persistent-Laplacian baseline on 13 targets, with `PBE0 atomization energy`, i.e., target T_1, being the single, pronounced exception.

Point-cloud experiments We evaluate `FiltSketch` ($T = 512$ and lags 1,4,16,64,256) on the point-cloud benchmarks, Flood-complex filtrations, and evaluation protocol of Graf et al. (2025): `Rocks` classification (2 classes), `MCB-C` (11 classes), `ModelNet10` (10 classes), and `Rocks` surface regression. For classification, we report mean test balanced accuracy and sample standard deviation across the ten prescribed splits; for regression, we report test mean squared error (MSE). We additionally compute persistence barcodes from the same Flood filtrations and compare with persistence images (Adams et al., 2017) followed by a ResNet-like CNN (He et al., 2016) with a number of parameters similar to that of the TCN, and ATOL (Royer et al., 2021) followed by XGBoost (Chen & Guestrin, 2016). For both PH baselines, we tune the vectorization hyperparameters (see Appendix C.1), and select the best configuration from the validation data within each split. These results therefore represent tuned PH-vectorization baselines rather than single default configurations.

`FiltSketch (Flood)` yields the strongest results on `Rocks` and `MCB-C`, but substantially lower accuracy on `ModelNet10` when taking neural baselines into account. Persistence images and ATOL

Table 2: Classification/Regression experiments. The **top** part contains comparisons to methods that leverage topological information, in particular PH combined with vectorization techniques (PIs and ATOL) tuned on a parameter grid. The **bottom** part compares to neural-network only baselines.

	ModelNet10	**MCB-C**	**Rocks** (clf)	**Rocks** (reg)
`FiltSketch (Flood)`	0.74 ± 0.02	0.77 ± 0.03	0.97 ± 0.01	$(0.3 \pm 0.1)\times$1e-3
`Flood PH + ATOL`	0.72 ± 0.02	0.74 ± 0.02	0.88 ± 0.03	$(0.5 \pm 0.2)\times$1e-3
`Flood PH + PI`	0.72 ± 0.01	0.68 ± 0.03	0.93 ± 0.02	$(1.4 \pm 0.1)\times$1e-3
`FiltSketch (Cubical)`	0.92 ± 0.04	–	–	–
PointNet++ (Qi et al., 2017)	0.93 ± 0.01	0.76 ± 0.02	0.54 ± 0.01	$(18.6 \pm 2.8)\times$1e-3
PointMLP (Ma et al., 2022)	0.93 ± 0.01	0.74 ± 0.03	0.51 ± 0.03	$(8.8 \pm 1.0)\times$1e-3
PVT (Zhang et al., 2021)	0.94 ± 0.01	0.77 ± 0.03	0.51 ± 0.04	$(31.4 \pm 3.4)\times$1e-3

built from the same filtration exhibit a similar limitation, suggesting that the principal bottleneck is the chosen filtration rather than the downstream representation alone. To test this hypothesis, we voxelize each `ModelNet10` shape as a 3D occupancy grid with voxel width 0.05 and construct cubical height filtrations from 26 directions, in the spirit of the persistent homology transform (Turner et al., 2014). We train `FiltSketch` using 4 subsets of these directional views, with `FiltSketch` representations aggregated before the TCN via a small DeepSet module. This change raises balanced accuracy to 0.92 ± 0.04 without changing the underlying filtration-sketch methodology. This contrast shows that matching the filtration and complex construction to the geometry of the task can be at least as important as the choice of topological vectorization.

Finally, to assess the relevance of each stream in `FiltSketch` and its sensitivity to the choice of the number of filtration levels, we further ran a sensitivity and an ablation experiment on `Rocks`.

Sensitivity to filtration sampling First, we vary the number of filtration levels (T) while retaining both representation streams and the preceding training and evaluation protocol. Table 3 shows that accuracy increases from 0.86 at 128 levels to 0.94 at 256 and 0.97 at 512. The smaller improvement from 256 to 512 suggests diminishing returns, while the overall trend indicates that sufficiently dense sampling is important for resolving changes along the filtration.

Table 3: Sampling sensitivity.

Resolution	Bal. accuracy
128	0.86 ± 0.03
256	0.94 ± 0.01
512	0.97 ± 0.01

Ablation of the two streams Next, we evaluate the Hodge and persistent-rank streams separately and in combination. For `Rocks`, we report test balanced accuracy. For `QM7b`, the reported ratio is the geometric mean across targets of the per-target MAE relative to Davies et al. (2023); hence, values below one indicate an aggregate improvement over that baseline. From Table 4, we see that the rank stream provides most of the predictive performance on both datasets, whereas the Hodge stream is substantially weaker in isolation. However, combining the streams improves the mean result over ranks alone in both cases, indicating that the Hodge stream contributes complementary information.

Table 4: Stream ablation.

`FiltSketch`	`Rocks` Bal. acc. ↑	`QM7b` MAE ratio ↓
`Hodge only`	0.83 ± 0.03	0.98
`Ranks only`	0.96 ± 0.01	0.90
`Hodge + Ranks`	0.97 ± 0.01	0.87

6 Discussion

We present filtration sketches as an alternative topology- and geometry-informed representation for learning. `FiltSketch` avoids explicit eigendecomposition and PH computation and our experiments establish the predictive utility of this construction. On various benchmarks, `FiltSketch` achieves higher mean performance than PH-based baselines. On `QM7b`, it also improves on an adapted TopoFormer baseline across all targets and on the persistent-Laplacian baseline for 13 of 14 targets. Our ablations suggest that persistent ranks provide the main predictive signal, while local Hodge summaries can improve the combined representation. Overall, our results support the conclusion that computing exact topological quantities might not be crucial to downstream performance, so long as this information can be captured approximately at a reasonable number of filtration levels.

Reproducibility statement

We provide the source code of `FiltSketch` as an anonymized archive in the supplementary material that covers (i) filtration construction, (ii) the randomized persistent-rank and Hodge feature computation, and (iii) model training and evaluation. In its readme-file, we provide example commands for the `QM7b` and `Rocks` regression experiments and a script that compiles the `QM7b` results. All datasets are publicly available, and we follow the evaluation protocols and splits of Davies et al. (2023) for `QM7b` and of Graf et al. (2025) for the point-cloud benchmarks. A single parametrization and training protocol is used across all experiments (Section 5). The randomized trace estimator and persistent-rank feature processing are specified in appendices A and B, derivations are given in Appendix C.2, and the computational complexity is analyzed in Appendix D.

AI use statement

In this work, we used generative AI tools to assist with implementing our method, specifically for writing parts of the source code. All AI-assisted code was reviewed and tested by the authors, who retained full control over the implementation. We also used generative AI tools to draft isolated paragraphs, to improve the presentation of existing derivations, and to edit the paper for grammar, spelling, and readability. All drafted text was revised by the authors, and every suggested edit was reviewed before being adopted. We have not used generative AI tools to generate data sets, develop theoretical models or the conceptual framework, formulate mathematical claims, provide ingredients for or write proofs, propose or refine hypotheses, design or provide feedback on the methodology or experiments, clean or reformat datasets, or interpret results; translation and qualitative data analysis are not applicable to this work. Generative AI tools were not used for literature search or reference management. All methodological and experimental decisions were made by the authors, and we take full responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

References

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

Haim Avron and Sivan Toledo. Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix. *Journal of the ACM*, 58(2):8:1–8:34, 2011. doi: 10.1145/1944345.1944349.

Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009. doi: 10.1090/S0273-0979-09-01249-X.

Mathieu Carrière, Frédéric Chazal, Yuichi Ike, Théo Lacombe, Martin Royer, and Yuhei Umeda. PersLay: A neural network layer for persistence diagrams and new graph topological signatures. In *AISTATS*, 2020.

Tianqi Chen and Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. In *SIGKDD*, 2016.

Thomas Davies, Zhengchao Wan, and Rubén J. Sánchez-García. The persistent laplacian for data science: Evaluating higher-order persistent spectral representations of data. In *ICML*, 2023.

Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. *Discrete & Computational Geometry*, 28(4):511–533, 2002. doi: 10.1007/s00454-002-2885-2.

Miroslav Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(2): 298–305, 1973.

Florian Graf, Paolo Pellizzoni, Martin Uray, Stefan Huber, and Roland Kwitt. The Flood complex: Large-scale persistent homology on millions of points. In *NeurIPS*, 2025.

Insu Han, Dmitry Malioutov, Haim Avron, and Jinwoo Shin. Approximating spectral sums of large-scale matrices using stochastic chebyshev approximations. *SIAM Journal on Scientific Computing*, 39(4):A1558–A1585, 2017. doi: 10.1137/16M1078148.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Christoph Hofer, Florian Graf, Bastian Rieck, Marc Niethammer, and Roland Kwitt. Graph filtration learning. In *ICML*, 2020.

Christoph D. Hofer, Roland Kwitt, and Marc Niethammer. Learning representations of persistence barcodes. *Journal of Machine Learning Research*, 20(126):1–45, 2019.

Danijela Horak and Jürgen Jost. Spectra of combinatorial Laplace operators on simplicial complexes. *Advances in Mathematics*, 244:303–336, 2013. doi: 10.1016/j.aim.2013.05.007.

Michael F. Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics—Simulation and Computation*, 18(3):1059–1076, 1989. doi: 10.1080/03610918908812806.

Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking network design and local geometry in point cloud: A simple residual MLP framework. In *ICLR*, 2022.

Facundo Mémoli, Zhengchao Wan, and Yusu Wang. Persistent laplacians: Properties, algorithms and implications. *SIAM Journal on Mathematics of Data Science*, 4(2):858–884, 2022. doi: 10.1137/21M1435471.

Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, 2017.

Martin Royer, Frédéric Chazal, Clément Levrard, Yuhei Umeda, and Yuichi Ike. ATOL: Measure vectorization for automatic topologically-oriented learning. In *AISTATS*, 2021.

Anirudh Som, Hongjun Choi, Karthikeyan Natesan Ramamurthy, Matthew P. Buman, and Pavan Turaga. PI-Net: A deep learning approach to extract topological persistence images. In *CVPR Workshops*, 2020.

Astrit Tola, Funmilola Mary Taiwo, Cuneyt Akcora, and Baris Coskunuzer. TopER: Topological embeddings in graph representation learning. In *NeurIPS*, 2025.

Katharine Turner, Sayan Mukherjee, and Doug M. Boyer. Persistent homology transform for modeling shapes and surfaces. *Information and Inference: A Journal of the IMA*, 3(4):310–344, 2014. doi: 10.1093/imaiai/iau011.

Md Joshem Uddin, Astrit Tola, Cuneyt Gurcan Akcora, and Baris Coskunuzer. TopoFormer: Topology meets attention for graph learning. In *ICLR*, 2026.

Zuoyu Yan, Tengfei Ma, Liangcai Gao, Zhi Tang, Yusu Wang, and Chao Chen. Neural approximation of graph topological features. In *NeurIPS*, 2022.

Cheng Zhang, Haochen Wan, Xinyi Shen, and Zizhao Wu. PVT: Point-voxel transformer for 3d deep learning. *arXiv preprint arXiv:2108.06076*, 2021. URL https://arxiv.org/abs/2108.06076.

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

Supplementary Material

A Randomized trace estimation

Let $\mathbf{M} \in \mathbb{R}^{n\times n}$ be a fixed symmetric matrix whose action on vectors can be evaluated without explicitly forming $\mathbf{M}$. Draw independent Rademacher probes $\mathbf{z}_1, \dots, \mathbf{z}_R \in \{-1, +1\}^n$ and define

$$\mathcal{T}_R(\mathbf{M}) := \frac{1}{R}\sum_{r=1}^{R} \mathbf{z}_r^\top \mathbf{M}\mathbf{z}_r. \tag{29}$$

Since $\mathbb{E}[\mathbf{z}_r\mathbf{z}_r^\top] = \mathbf{I}_n$, we have $\mathbb{E}[\mathcal{T}_R(\mathbf{M})] = \mathrm{tr}(\mathbf{M})$. This identity does not require positive semidefiniteness (Hutchinson, 1989; Avron & Toledo, 2011). For a polynomial g and a symmetric operator $\mathbf{A}$, we may therefore estimate $\mathrm{tr}(g(\mathbf{A}))$ by applying Eq. (29) to $\mathbf{M} = g(\mathbf{A})$.

B Persistent-rank feature processing

Before constructing the feature channels, we apply a numerical safeguard to the restricted Gram trace estimate. For $N_{q+1} > 0$, define

$$\widehat{u}_{q+1,k}(\mathcal{K} \hookrightarrow \mathcal{K}') := \frac{1}{N_{q+1}} \mathcal{T}_R\big((\mathbf{I} - \mathbf{G}_{q+1})^k\big) \quad . \tag{30}$$

In the implementation, we evaluate the restricted-rank term as

$$\widehat{a}_{q+1,k}(\mathcal{K} \hookrightarrow \mathcal{K}') := N_{q+1}\left(1 - \mathrm{clip}_{[0,1]}\big(\widehat{u}_{q+1,k}(\mathcal{K} \hookrightarrow \mathcal{K}')\big)\right) \quad , \tag{31}$$

where $\mathrm{clip}_{[a,b]}(x) := \min\{b, \max\{a, x\}\}$. For $N_{q+1} = 0$, set $\widehat{a}_{q+1,k}(\mathcal{K} \hookrightarrow \mathcal{K}') = 0$. The clipping is inactive in exact arithmetic because the diffusion power is a positive semidefinite contraction and each Rademacher probe has squared norm N_{q+1}.

The feature actually supplied to the model is

$$\psi\big(\widehat{\beta}_{q,k}(\mathcal{K} \hookrightarrow \mathcal{K}')\big) := \log\Big(1 + \mathrm{clip}_{[0,\max\{0,\, n_q(\mathcal{K})\, \widehat{m}^{\mathrm{down}}_{q,k}(\mathcal{K})\}]}\big(\widehat{\beta}_{q,k}(\mathcal{K} \hookrightarrow \mathcal{K}')\big)\Big). \tag{32}$$

This second clipping operation imposes nonnegativity and an estimated source-cycle cap. It can be active even in exact arithmetic because $\widehat{\beta}_{q,k}$ combines approximate terms. The logarithm compresses the count-dependent dynamic range. The estimated cap need not contain the exact persistent rank, and the processed features are not unbiased rank estimates. For unavailable filtration level pairs with $s + \ell > T$, the features are set to zero.

C Additional details

C.1 Hyperparameter selection for PH vectorization baselines

For both PH baselines, we compute persistence barcodes of H_0, H_1 and H_2 from the same filtrations used by `FiltSketch`. Within each split, we select the configuration with the best validation performance and evaluate it once on the test data.

For persistence images (Adams et al., 2017), we search over resolutions $\{16, 32, 64, 128\}$ and five different Gaussian bandwidths. For ATOL (Royer et al., 2021), we search over $\{32, 64, 128\}$ centers and a stretch of $\{1\%, 5\%, 10\%\}$.

C.2 Establishing Eq. (16)

To establish Eq. (16), i.e.,

$$\beta_q(\mathcal{K} \hookrightarrow \mathcal{K}') = \dim \ker(\mathbf{B}_q(\mathcal{K})) - \dim(\mathrm{im}(\mathbf{B}_{q+1}(\mathcal{K}')) \cap C_q(\mathcal{K}; \mathbb{R})) \quad , \tag{33}$$

we first observe that a source cycle $\gamma \in \ker \mathbf{B}_q(\mathcal{K})$ maps to zero in the target homology precisely when it is a target boundary, i.e., $\gamma \in \mathrm{im}\, \mathbf{B}_{q+1}(\mathcal{K}')$.

Thus,

$$\beta_q(\mathcal{K} \hookrightarrow \mathcal{K}') = \dim \ker \mathbf{B}_q(\mathcal{K}) - \dim \left(\ker \mathbf{B}_q(\mathcal{K}) \cap \mathrm{im}\, \mathbf{B}_{q+1}(\mathcal{K}')\right) \quad . \tag{34}$$

Moreover, we have

$$\ker \mathbf{B}_q(\mathcal{K}) \cap \mathrm{im}\, \mathbf{B}_{q+1}(\mathcal{K}') = C_q(\mathcal{K}; \mathbb{R}) \cap \mathrm{im}\, \mathbf{B}_{q+1}(\mathcal{K}') \quad . \tag{35}$$

First, since $\ker \mathbf{B}_q(\mathcal{K}) \subseteq C_q(\mathcal{K}; \mathbb{R})$, we have

$$\ker \mathbf{B}_q(\mathcal{K}) \cap \mathrm{im}\, \mathbf{B}_{q+1}(\mathcal{K}') \subseteq C_q(\mathcal{K}; \mathbb{R}) \cap \mathrm{im}\, \mathbf{B}_{q+1}(\mathcal{K}') \quad . \tag{36}$$

Second, since boundaries are cycles, we have

$$\mathrm{im}\, \mathbf{B}_{q+1}(\mathcal{K}') \cap C_q(\mathcal{K}; \mathbb{R}) \subseteq \ker \mathbf{B}_q(\mathcal{K}') \cap C_q(\mathcal{K}; \mathbb{R}) \quad . \tag{37}$$

Moreover, since the source and target boundary maps agree on source chains, we have

$$\ker \mathbf{B}_q(\mathcal{K}') \cap C_q(\mathcal{K}; \mathbb{R}) \subseteq \ker \mathbf{B}_q(\mathcal{K}) \quad , \tag{38}$$

and thus,

$$\mathrm{im}\, \mathbf{B}_{q+1}(\mathcal{K}') \cap C_q(\mathcal{K}; \mathbb{R}) \subseteq \ker \mathbf{B}_q(\mathcal{K}) \cap \mathrm{im}\, \mathbf{B}_{q+1}(\mathcal{K}') \quad . \tag{39}$$

D Computational complexity

Consider one filtration with S simplices and E nonzero boundary-matrix entries, summed over the chain dimensions used by the sketches.

Let $K = \max \mathcal{P}$ be the largest diffusion depth and let $M = \sum_{\ell \in \mathcal{L}} \max\{T - \ell, 0\}$ be the number of queried filtration level pairs. We store the sparse boundary matrices once and mask them for each sampled complex or pair.

At each diffusion step, the lower, upper, and full Hodge filters require a constant number of sparse boundary products with R probe vectors. Constructing the Hodge stream therefore costs $O(RKT(S+E))$. The persistent-rank stream reuses the lower-Hodge estimates for its source-cycle and target-boundary terms; evaluating the restricted Gram term requires additional masked boundary products for each queried pair, costing $O(RKM(S+E))$. Hence, sketch construction is bounded by

$$O(RK(T+M)(S+E)) \quad . \tag{40}$$

For a fixed number of lags, $M = O(T)$, and the bound is linear in T, R, K, and the size of the stored boundary representation.

Notably, these bounds exclude construction of the underlying filtration and training of the sequence model.