The Persistent Laplacian for Data Science: Evaluating Higher-Order Persistent Spectral Representations of Data

Thomas Davies [1 2] **Zhengchao Wan** [3] **Ruben Sanchez-Garcia** [1 2]

Abstract

Persistent homology is arguably the most successful technique in Topological Data Analysis. It combines homology, a topological feature of a data set, with persistence, which tracks the evolution of homology over different scales. The persistent Laplacian is a recent theoretical development that combines persistence with the combinatorial Laplacian, the higher-order extension of the well-known graph Laplacian. Crucially, the Laplacian encodes both the homology of a data set, and some additional geometric information not captured by the homology. Here, we provide the first investigation into the efficacy of the persistent Laplacian as an embedding of data for downstream classification and regression tasks. We extend the persistent Laplacian to cubical complexes so it can be used on images, then evaluate its performance as an embedding method on the MNIST and MoleculeNet datasets, demonstrating that it consistently outperforms persistent homology.

1. Introduction

Topological Data Analysis (TDA) encompasses a group of techniques that use topologically-inspired methods in data analysis (Edelsbrunner & Harer, 2022; Dey & Wang, 2022). Topology studies the same objects as geometry ('shapes') but allows extreme deformations such as stretching, twisting, or bending, and thus it is concerned with properties (called 'topological invariants') that remain unchanged under such deformations. Although we should think of 'shapes' as continuous objects, we can discretise them without losing any topological information, and, indeed, many topological invariants are calculated directly on these discretizations. Using these, we can convert 'data' into 'shape', by constructing one of these discretizations from a data set (typically a simplicial or cubical complex), and then extracting topological features – this is the standard TDA workflow.

A key idea behind many TDA techniques is that of persistence. Instead of counting a topological feature (e.g. number of connected pieces), we track its evolution on a family of increasing discretizations, called a filtration, that encodes a 'shape', or data set, at different 'scales'. The combination of persistence with homology, a powerful topological invariant, is by far the most successful TDA technique in applications across a variety of domains (see e.g., Bukkuri et al. (2021) and Pritchard et al. (2022)).

A natural extension of this approach, moving towards what one could call Geometric Data Analysis, would be to incorporate more geometrical information missed by homology, that is, to supplement homology, a purely topological invariant, with some geometrical information of the underlying data set. The combinatorial, or discrete, Laplacian operator (Horak & Jost, 2013), a higher-order generalization of the well-known graph Laplacian (Eckmann, 1944; Goldberg, 2002), is the perfect candidate for this task. It incorporates the homology, as its 0-eigenspace, while the non-zero spectrum captures aspects of the geometry. As an illustration, the 0-eigenvalues of the graph Laplacian correspond to the connected components of a graph (its 0-homology, a topological invariant), while the non-zero eigenvalues relate to clusters, or 'almost connected components', a geometric feature undetected by homology, and the basis of the spectral clustering algorithms (Von Luxburg, 2007).

The persistent Laplacian, a recent theoretical development (Lieutier, 2014; Wang et al., 2020; Mémoli et al., 2022), extends the higher-order Laplacian operator to the persistence setting. In particular, the 0-eigenvalues of the persistent Laplacians recover information about the persistent homology (Mémoli et al., 2022). This gives a powerful new way to summarise data that combines topological and geometric information with the persistence TDA hallmark.

In this article, we introduce the persistent Laplacian to the Machine Learning community as a feature vector which

[1]University of Southampton, UK [2]The Alan Turing Institute, The British Library, UK [3]Halıcıoğlu Data Science Institute, University of California San Diego, La Jolla, California, USA. Correspondence to: Thomas Davies <tdavies@turing.ac.uk>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

encodes persistent homology plus additional aspects of the underlying geometry, and assess its usefulness in practical applications. Namely, we empirically investigate its efficacy in comparison to other topological and geometric techniques at MNIST classification and the prediction of molecular properties from the MoleculeNet dataset (Wu et al., 2018). Additionally, we evaluate the importance of utilising the filtration in the persistent Laplacian and different magnitude eigenvalues to the success on the downstream task. Our code is available on GitHub.[1]

As far as we know, this article is the first empirical evidence that the persistent Laplacians outperforms the current mainstream topological tools on real-life datasets, paving the way for its wider use by the ML and TDA communities.

1.1. Related Work

The theory for the persistent Laplacian has only recently been developed, and we believe we provide the first evaluation of the persistent Laplacian as an applied tool. The (non-persistent) combinatorial Laplacian can be used as an update step for graph neural networks on simplicial complexes (Bodnar et al., 2021), analogously to how the graph Laplacian is used as an update step for graph convolutional networks (Kipf & Welling, 2017). It is also used to analyze protein datasets by Wang et al. (2020; 2021); Meng & Xia (2021); Wee & Xia (2022), chromosomal structure in Gong et al. (2022), and for drug design applications in Jiang et al. (2022). In each of these applications the authors compute the combinatorial Laplacian over a filtration, rather than the persistent Laplacian, as they do not define the Laplacian via the persistent boundary operator and related chain groups (see Section 2.5 for details). An implementation of the persistent Laplacian was provided in Matlab by Mémoli et al. (2022), but was not used for data analysis applications in their paper.

The work in Qiu & Wei (2023) similarly investigates spectral techniques, linking their methods back to the persistent Laplacian. This work is distinct from ours as although they demonstrate how their methods are related to the persistent Laplacian, they do not use it as a feature vector. In particular, when using the persistent Laplacian the authors ‘only extracted features from harmonic spectra of persistent Laplacians coding topological invariants for the high-dimensional interactions’. This corresponds exactly to persistent homology by Mémoli et al. (2022, Theorem 2.7).

1.2. Our Contributions

The purpose of this paper is to introduce the persistent Laplacian to practitioners within TDA and, more broadly, those interested in encoding structure in a data set using topological and geometrical features. Alongside accessible explanations of the necessary theoretical background, our research contributions demonstrate its added value in practical Machine Learning applications, compared to purely topological methods. The three primary contributions of this paper are as follows.

(i) **Extension to cubical complexes.** We extend the theory behind the persistent Laplacian from simplicial complexes (original setting) to cubical complexes, making these methods immediately applicable to image data sets.

(ii) **Baselines.** We evaluate the persistent Laplacian spectrum as a feature vector and find that it consistently outperforms persistent homology across digit recognition and molecular property prediction tasks. It also equals or outperforms its non-persistent versions, the graph and combinatorial Laplacians.

(iii) **Python implementation.** We provide the first implementation of the persistent Laplacian in Python. We have released a full version of the codebase on GitHub, making the persistent Laplacian readily available to non-specialists working as data science practitioners.

1.3. Paper Structure

In Section 2, we introduce the necessary theory behind the persistent Laplacian, including how it relates, and supersedes, the graph Laplacian and its higher-dimensional generalization, the combinatorial Laplacian. In Section 3, we introduce cubical complexes, explain how to efficiently compute the persistent Laplacian using the Schur complement, and extend this computation to the cubical case. In Section 4, we build intuition for the persistent Laplacian eigenvalues, using simple synthetic data, and MNIST, and illustrate how we compute and represent the persistent Laplacian as a feature vector. Finally, in Section 5, we run experiments on MNIST and MoleculeNet, using persistent homology, the graph Laplacian, and the combinatorial Laplacian as baselines.

2. Theoretical Background

2.1. Simplicial Complexes

A key data structure for both persistent homology and the Laplacian is the simplicial complex. A *simplicial complex* is a collection of p-dimensional triangles, called p-*simplices*, i.e., vertices, edges, triangles, tetrahedrons, and so on. Note that if we just consider the first two dimensions, we have vertices (0-simplices) and edges (1-simplices), resulting in a graph, so we can view simplicial complexes as a generalization of a graph that can represent higher-order interactions

[1] `https://github.com/tomogwen/persistentlaplaciandatascience`

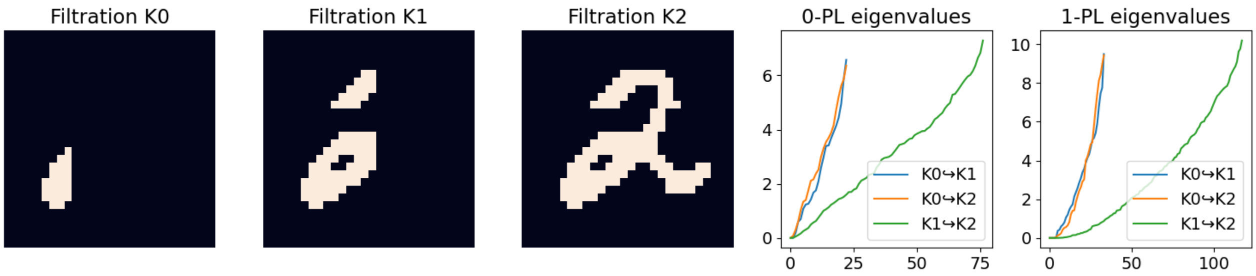


Figure 1: A filtration of cubical complexes $K_0 \hookrightarrow K_1 \hookrightarrow K_2$ of an MNIST digit, alongside the eigenvalues of the persistent Laplacian. The x axis is the eigenvalue position and the y axis is its value (i.e., the y value at $x = a$ is the value of the ath smallest eigenvalue).

and structures in data (p-simplices corresponding to $(p+1)$-cliques in the graph). Beyond extending graphs, simplicial complexes provide a natural discretization of continuous 'shapes' (e.g. a tessellation of a surface into triangles) and data sets (with data points as vertices). We identify a simplex with its set of vertices, and call it *oriented* if we fix an *orientation*, that is, an ordering of its vertices. We write $\{a\}$, $\{a,b\}$, $\{a,b,c\}$, etc, respectively $[a]$, $[a,b]$, $[a,b,c]$ etc, for vertices, edges, triangles etc, respectively oriented vertices, edges, triangles etc (so that $\{a,b\} = \{b,a\}$ but $[a,b] \neq [b,a]$, for example).

2.2. Boundary Operators and Homology

Given a simplicial complex K, the (oriented) p-dimensional simplices can be viewed as generators for a vector space called the *p-chain group* C_p^K, made of linear combinations of p-simplices. The *boundary operator* on these chain groups is a linear map $\partial_p^K : C_p^K \to C_{p-1}^K$ that sends a p-dimensional simplex to its $p-1$ boundary. For example, it sends a triangle to its constituent edges (namely, $\partial_2([a,b,c]) = [b,c] - [a,c] + [a,b]$, a linear combination of its 3 edges). The boundary operator corresponds to multiplication by the *boundary matrix* B_p^K, whose (i,j)-entry is ± 1 if the jth (oriented) $(p-1)$-simplex is a constituent of the ith (oriented) p-simplex, and 0 otherwise, with sign depending on orientations. All in all, we are able to convert purely topological information (simplices and boundary relations) into purely algebraic information (vector spaces and linear maps).

The chain groups and boundary operators form a *chain complex*,

$$\cdots \xrightarrow{\partial_{p+2}^K} C_{p+1}^K \xrightarrow{\partial_{p+1}^K} C_p^K \xrightarrow{\partial_p^K} C_{p-1}^K \xrightarrow{\partial_{p-1}^K} \cdots,$$

that is, a sequence of linear maps satisfying $\partial_p^K \circ \partial_{p+1}^K = 0$ (this can be shown from the choice of orientation signs in the definition of boundary operator). In particular, an element in the image of one boundary map, is in the kernel of the next one, and we can define the *homology groups* $H_p^K = Z_p^K / B_{p+1}^K$, where $Z_p^K = \ker \partial_p^K$ and $B_{p+1}^K = \operatorname{im} \partial_{p+1}^K$. Although not obvious from the definition, these homology groups capture important topological properties of the simplicial complex K. For instance, their rank, called the Betti number $\beta_p^K = \operatorname{rank} H_p^K$, captures the number of connected components for $k = 0$, the number of holes for $k = 1$, the number of voids for $k = 2$, and so on.

2.3. Persistent Homology

Simplicial complexes can be constructed from data in a number of ways, but typically by a parameterized method like the *Vietoris-Rips complex*, which connects $k+1$ points into a k-simplex when they are pairwise within ϵ distance of each other. The choice of parameter ϵ varies the topological properties of the simplicial complex. The idea behind *persistence* is to allow all choices simultaneously, that is, to consider a family of increasing simplicial complexes, called a filtration, and track the evolution of homology over the filtration. Formally, we define a *simplicial pair*, written $K \subseteq L$, as two simplicial complexes K and L with the same set of vertices and such that all simplices of K are simplices of L. Then, a *filtration* is a family $\{K_t\}_{t \in \mathbb{R}}$ of simplicial complexes with a simplicial pair $K_t \subseteq K_{t'}$ whenever $t \leq t'$.

If $K \subseteq L$, the boundary maps of K and of L can be written as a commutative diagram where vertical arrows are inclusion maps

$$\begin{tikzcd}
\cdots \arrow[r] & C_{p+1}^K \arrow[r, "\partial_{p+1}^K"] \arrow[d, hook, dashed] & C_p^K \arrow[r, "\partial_p^K"] \arrow[d, hook, dashed] & C_{p-1}^K \arrow[r] \arrow[d, hook, dashed] & \cdots \\
\cdots \arrow[r] & C_{p+1}^L \arrow[r, "\partial_{p+1}^L"] & C_p^L \arrow[r, "\partial_p^L"] & C_{p-1}^L \arrow[r] & \cdots
\end{tikzcd}$$

Now we can define *persistent homology* on a simplicial complex pair $K \hookrightarrow L$ as $H_p^{K,L} = Z_p^K / (B_p^L \cap Z_p^K)$. This represents topological features present in K that persist to

L, i.e., are still present (non-zero) in L. We can similarly define the persistent Betti number $\beta_p^{K,L} = \operatorname{rank} H_p^{K,L}$, which counts topological features persisting from K to L.

2.4. The Graph and Combinatorial Laplacian

The graph Laplacian can also be seen in terms of the boundary operator, which makes the connection to homology clearer. If G is a graph, the boundary operator $\partial_1 \colon C_1 \to C_0$ sends an oriented edge $[a, b]$ to $[b] - [a]$. In matrix form, this operator corresponds to multiplication by the incidence matrix B of the oriented graph (the graph with an arbitrary, but fixed, orientation of the edges), that is, the matrix with (i, j)-entry 1, respectively -1, if the jth vertex is the source, respectively target, of the ith edge. We have a dual map in the other direction, $(\partial_1)^* \colon C_0 \to C_1$, called the *coboundary operator*, which corresponds simply to multiplication by the transpose matrix B^T. The graph Laplacian operator Δ_0 is then defined as the composition of the boundary and coboundary operators, $\Delta_0 \colon C_0 \to C_0$, $\Delta_0 = \partial_1 \circ (\partial_1)^*$. In matrix terms, this corresponds to the matrix BB^T, which equals $D - A$ (regardless of the chosen edge orientations), where A is the adjacency matrix of the graph, and D the diagonal matrix of vertex degrees, recovering the standard definition of the graph Laplacian matrix.

The above interpretation in terms of boundary and coboundary operators give a straightforward definition of Laplacian for arbitrary simplicial complexes (see (Horak & Jost, 2013) for details). Consider a simplicial complex K, and the boundary and coboundary operators at each dimension,

$$\begin{tikzcd}
\cdots \arrow[r] & C^K_{p+1} \arrow[r, shift right, "\partial^K_{p+1}"'] & C^K_p \arrow[l, shift right, "(\partial^K_{p+1})^*"'] \arrow[r, shift right, "\partial^K_p"'] & C^K_{p-1} \arrow[l, shift right, "(\partial^K_p)^*"'] \arrow[r] & \cdots
\end{tikzcd}$$

We define the *up* and *down combinatorial Laplacians* as the linear maps

$$\Delta^K_{p,\mathrm{up}} = \partial^K_{p+1} \circ (\partial^K_{p+1})^* \quad \text{and} \quad \Delta^K_{p,\mathrm{down}} = (\partial^K_p)^* \circ \partial^K_p$$

respectively, and the *combinatorial Laplacian* as $\Delta^K_p = \Delta^K_{p,\mathrm{up}} + \Delta^K_{p,\mathrm{down}}$. For a graph, the up 0-Laplacian coincides with the graph Laplacian (see above), and the down 0-Laplacian is zero. In terms of matrix representations, the combinatorial Laplacian Δ^K_p corresponds to the matrix $B^K_{p+1} \cdot (B^K_{p+1})^T + (B^K_p)^T \cdot B^K_p$, where B^K_p etc are the boundary matrices described in Section 2.2.

Moreover, the combinatorial Laplacian captures the non-persistent homology at each dimension. Explicitly, one can show that $H^K_p = \ker(\Delta^k_p)$, a discrete version of a classical result in Hodge Theory (see Horak & Jost (2013)). The non-zero spectrum of the combinatorial Laplacian is less well-understood and one goal of this article is to shed light on its significance.

2.5. The Persistent Laplacian

Lieutier (2014) and Wang et al. (2020) independently developed the persistent Laplacian, a persistent version of the combinatorial Laplacian which makes the combinatorial Laplacian applicable to scenarios (filtrations of simplicial complexes) that were previously the sole domain of persistent homology. Recent work by Mémoli et al. (2022) established various theoretical properties of the persistent Laplacian as well as efficient algorithms for computing matrix representations of persistent Laplacians. We will now introduce a definition of the persistent Laplacian following the notation from Mémoli et al. (2022).

Let $K \subseteq L$ be a simplicial pair and consider the subspace of C^L_p given by

$$C^{L,K}_p = \left\{ c \in C^L_p \ : \ \partial^L_p(c) \in C^K_{p-1} \right\} \subseteq C^L_p. \tag{1}$$

That is, $C^{L,K}_p$ consists of all simplices in the larger simplicial complex, that is, in C^L_p, that have their boundary in the smaller simplicial complex, that is, in $C^K_{p-1} \subseteq C^L_{p-1}$. Write $\partial^{L,K}_p$ for the restriction to this subspace $\partial^L_p|_{C^{L,K}_p}$. Then, the *p-persistent Laplacian* is defined by

$$\Delta^{K,L}_p = \partial^{L,K}_{p+1} \circ \left(\partial^{L,K}_{p+1}\right)^* + (\partial^K_p)^* \circ \partial^K_p, \tag{2}$$

with the *up p-persistent Laplacian* given by $\Delta^{K,L}_{p,\mathrm{up}} = \partial^{L,K}_{p+1} \circ \left(\partial^{L,K}_{p+1}\right)^*$. The relation between each operator is shown in the diagram from Mémoli et al. (2022) below.

$$\begin{tikzcd}
C^K_{p+1} \arrow[rr, "\partial^K_{p+1}"] \arrow[dd, hook, dashed] & & C^K_p \arrow[dl, blue, shift right, "\left(\partial^{L,K}_{p+1}\right)^*"'] \arrow[r, blue, shift right, "\partial^K_p"'] \arrow[dd, hook, dashed] & C^K_{p-1} \arrow[l, blue, shift right, "(\partial^K_p)^*"'] \arrow[dd, hook, dashed] \\
& C^{L,K}_{p+1} \arrow[ur, blue, shift right, "\partial^{L,K}_p"'] \arrow[dl, hook, dashed] & & \\
C^L_{p+1} \arrow[rr, "\partial^L_{p+1}"] & & C^L_p \arrow[r, "\partial^L_p"] & C^L_{p-1}
\end{tikzcd}$$

A key property of the persistent Laplacian is that the number of zero eigenvalues of $\Delta^{K,L}_p$ is equal to $\beta^{K,L}_p$, the p-th persistent Betti number (cf. Theorem 2.7 in Mémoli et al. (2022)). That is, the kernel of the persistent Laplacian captures the rank of the p-th persistent homology groups. As mentioned already in Mémoli et al. (2022, Example 2.3), when K and L are both graphs with the same vertex set, then $\Delta^{K,L}_0$ reduces to the usual graph Laplacian Δ^L_0 of the larger graph L.

Despite the intimidating operator definition in Equation (2), a succinct matrix representation of the persistent Laplacian can be obtained via the Schur complement as we now explain (Mémoli et al., 2022).

Consider a block matrix $M = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \in \mathbb{R}^{n\times n}$ where D is a $d \times d$ matrix. The *(generalized) Schur complement* of D in M, denoted by M/D, is defined as $M/D := A - BD^{\dagger}C$, where $D^{\dagger}$ denotes the Moore-Penrose pseudoinverse of D.

In Mémoli et al. (2022), it is shown that the matrix representation for up persistent Laplacians of simplicial pairs can be computed via Schur complement of certain matrices. More precisely, let $K \hookrightarrow L$ be a simplicial pair, write $\mathbf{\Delta}^{L}_{\mathrm{up},p}$ for the matrix representation of the p-th up Laplacian $\Delta^{L}_{\mathrm{up},p}$ of L, and let I^{L}_{K} be the submatrix of $\mathbf{\Delta}^{L}_{\mathrm{up},p}$ with rows (or columns) corresponding to p-simplices not belonging to K. Then $\mathbf{\Delta}^{K,L}_{\mathrm{up},p} := \mathbf{\Delta}^{L}_{\mathrm{up},p}/\mathbf{\Delta}^{L}_{\mathrm{up},p}(I^{L}_{K}, I^{L}_{K})$.

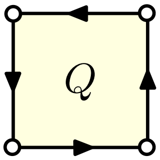


Figure 2: The boundary of Q is a signed sum of the geometric boundary edges of the square. Formally, $\partial_2([n,n+1]\times[m,m+1]) = [n+1]\times[m,m+1] + [n]\times[m,m+1] - [n,n+1]\times[m+1] - [n,n+1]\times[m]$ the sum of the vertical edges minus the horizontal edges.

3. Extension to Cubical Complexes

Just as a simplicial complex is built using simplices, a cubical complex is an analogue whose building blocks are cubes. Cubical complexes are useful for representing 2D/3D models in computer graphics, as pixels and voxels are easier to represented using squares and cubes rather than triangles and tetrahedra. They have also been studied in the field of persistent homology (Strömbom, 2007). We briefly recall some notions related to cubical complexes below and then establish how the persistent Laplacian theory can be applied to study cubical complexes.

Intervals in $\mathbb{R}$ of the form $[m, m+1]$ or $[m, m]$ for $m \in \mathbb{N}$ are called *elementary intervals*. $[m, m+1]$ is also called a *1-cube* whereas $[m, m]$ is called a degenerate elementary interval or a 0-*cube*. In $\mathbb{R}^n$, *elementary cubes* or n-*cubes* Q are defined to be products of n elementary intervals $I_1, \ldots, I_n$, i.e., $Q = I_1 \times \cdots \times I_n$. We let $\dim(Q)$ denote the number of non-degenerate components I_i. A subset $K \subseteq \mathbb{R}^n$ is called a *cubical complex* if it is the finite union of n-cubes.

One can define chain groups for K for each $k \in \mathbb{N}$: C^K_k is generated by all k-cubes Q with $\dim(Q) = k$. We endow an inner product on C^K_k such that the set of k-cubes is an orthonormal basis. We also define boundary maps $\partial_k : C^K_k \to C^K_{k-1}$, as follows. For a 0-cube $[m] = [m, m]$, its boundary $\partial_0([m])$ is defined as0. For a 1-cube $[m, m+1]$, the boundary is defined as $\partial_1([m, m+1]) := [m+1] - [m]$. Now, given $k > 1$ and a k-cube $Q = I_1 \times \cdots \times I_k$ of dimension k, its boundary is recursively defined as

$$\partial_k Q := \partial_1(I_1)\times I_2\times\cdots\times I_k + (-1)^{\dim(I_1)} I_1 \times \partial_{k-1}(I_2\times\cdots\times I_k).$$

(See Figure 2 for an illustration.)

Following this definition, it is easy to check that $\partial_k \circ \partial_{k+1} = 0$ for any $k \in \mathbb{N}$ (see for example (Strömbom, 2007) for more details) and, in particular, a cubical complex K gives rise to a chain complex

$$\cdots \xrightarrow{\partial^K_{p+2}} C^K_{p+1} \xrightarrow{\partial^K_{p+1}} C^K_p \xrightarrow{\partial^K_p} C^K_{p-1} \xrightarrow{\partial^K_{p-1}} \cdots$$

In this way, one can also develop the Laplacian theory for cubical complexes as already done in Duval et al. (2011). Similarly, we develop a persistent version of the Laplacian theory for cubical complexes. Given two cubical complexes K and L with $K \subseteq L$, we call them a *cubical pair*. As in the case of simplicial complexes, we define the subspace $C^{L,K}_p$ as in Equation (1), and the (up) persistent Laplacian $\Delta^{K,L}_p$ on C^K_p as in Equation (2).

We similarly extend the notation and terminology to the case of cubical pair $K \hookrightarrow L$ and, in particular, we can show that $\mathbf{\Delta}^{K,L}_{\mathrm{up},p} := \mathbf{\Delta}^{L}_{\mathrm{up},p}/\mathbf{\Delta}^{L}_{\mathrm{up},p}(I^{L}_{K}, I^{L}_{K})$ is the matrix representation of $\Delta^{K,L}_{\mathrm{up},p}$. This can be proved by directly adapting the proof of Theorem 4.6 in Mémoli et al. (2022), although it also follows from a more general result in Gülen et al. (2023) (see Appendix A for details).

4. The Persistent Laplacian as a Feature Vector

We have explained the theory behind the persistent Laplacian, and shown that we can also use it with cubical complexes, but it remains to be seen it is best utilized as a feature vector use in downstream tasks. Typically in Topological Data Analysis we will be working with a filtration: a collection of complexes K_i such that $K_i \subseteq K_j$, with i a continuous real parameter in some range $[i_0, i_T]$. Algorithms from persistent homology theory can handle the entire filtration efficiently and generate the so-called ‘persistence diagram’ (or ‘barcodes’) as the topological summary of the filtration, enabling us to understand the whole filtration via the births and deaths of topological features (Edelsbrunner et al., 2002). On the other hand, the persistent Laplacian $\Delta^{K,L}_p$ is defined for just two complexes: a complex pair (either a simplicial pair or a cubical pair) $K \hookrightarrow L$. The first question we must answer is given a filtration, which complex pairs we select to compute the persistent Laplacian. The second question is which dimensions of the persistent Laplacian to compute - in our investigations, we will typically compute

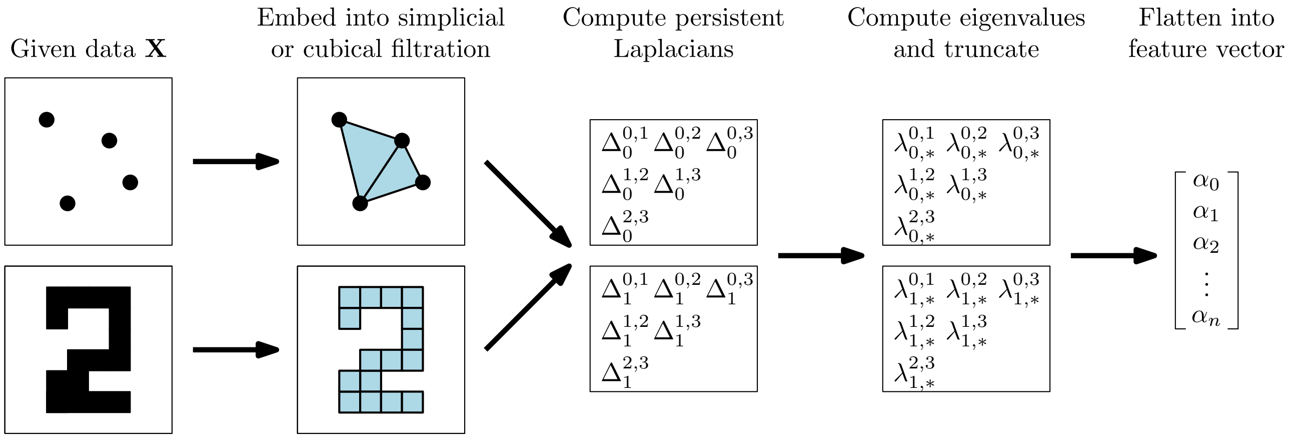

Given data **X**
Embed into simplicial or cubical filtration
Compute persistent Laplacians
Compute eigenvalues and truncate
Flatten into feature vector
$\Delta_0^{0,1}$ $\Delta_0^{0,2}$ $\Delta_0^{0,3}$
$\Delta_0^{1,2}$ $\Delta_0^{1,3}$
$\Delta_0^{2,3}$
$\Delta_1^{0,1}$ $\Delta_1^{0,2}$ $\Delta_1^{0,3}$
$\Delta_1^{1,2}$ $\Delta_1^{1,3}$
$\Delta_1^{2,3}$
$\lambda_{0,*}^{0,1}$ $\lambda_{0,*}^{0,2}$ $\lambda_{0,*}^{0,3}$
$\lambda_{0,*}^{1,2}$ $\lambda_{0,*}^{1,3}$
$\lambda_{0,*}^{2,3}$
$\lambda_{1,*}^{0,1}$ $\lambda_{1,*}^{0,2}$ $\lambda_{1,*}^{0,3}$
$\lambda_{1,*}^{1,2}$ $\lambda_{1,*}^{1,3}$
$\lambda_{1,*}^{2,3}$
α_0 α_1 α_2 $\vdots$ α_n

Figure 3: The pipeline to compute feature vectors from data using the persistent Laplacian. We first embed the data into a filtration of cubical or simplicial complexes, then compute the persistent Laplacian for a selection of complex pairs and dimensions, before taking the eigenvalues, truncating/zero-padding (if necessary), then flattening into a feature vector.

the first two or three dimensions of the persistent Laplacian, as is commonly done in TDA.

To answer the first question, we point out that the time complexity for computing the persistent Laplacian for a given simplicial pair $K \hookrightarrow L$ is similar to (and sometimes faster than) the one for computing the persistent homology for the pair (Mémoli et al., 2022). However, computing the persistent Laplacian for each pair of simplicial complexes inside a filtration is time consuming. In particular, consider a simplex-wise filtration $K_1 \subseteq \cdots \subseteq K_N$, i.e., K_i and K_{i+1} differ by only one simplex for $i = 1, \ldots, N-1$. Assume further that K_1 is just a single vertex, so that N coincides with the total number of simplices in K_N. Let $q := \dim(K_N)$ and let n_p denote the number of p-simplices in K_N. By Theorem 5.1 in Mémoli et al. (2022), given any $p = 0, \ldots, q$, computing $\{\Delta_p^{i,j}\}_{1\le i\le j\le N}$ takes time $O\left(N^2(n_p)^2 + Nn_{p+1}\right)$. Then, to compute $\{\Delta_p^{i,j}\}_{1\le i\le j\le N, 0\le p\le q}$, one needs time

$$O\left(N^2 \sum_{p=0}^{q} n_p^2 + N \sum_{p=0}^{q} n_{p+1}\right) = O\left(N^2 \sum_{p=0}^{q} n_p^2\right).$$

Note that $\sum_p n_p^2 \geq (\sum_p n_p)^2/(q+1) = N^2/(q+1)$. Hence, the total time complexity is $O(N^4/(q+1))$ which is asymptotically larger than $O(N^3)$, an upper bound for computing persistence diagrams of all dimensions using the standard persistent homology algorithm.

Based on the discussion above, we compute the persistent Laplacian only for pairs in a small subset of the filtration. We first select a resolution R (in practice, we found R = 4 to be effective). Then, we compute an increment $I = (i_T - i_0)/R$ to obtain a subset of the filtration $F = [i_0, i_0 + I, i_0 + 2I, \ldots, i_0 + RI = i_T]$. We then consider complex pairs $K_i \hookrightarrow K_j$ where $i, j \in F$ and $i \leq j$. An example of a sequence of complexes is given in Figure 1, where we show three sequential cubical complexes generated from an MNIST digit. Given the complex pairs, we compute their Laplacian and corresponding eigenvalues using our codebase. In Figure 1 we compute the persistent Laplacian for the three inclusion pairs generated from the subset of the filtration parameter $F = [0, 1, 2]$ and for dimension $p = 0, 1$. The plots reveal distinct differences among the eigenvalues.

To maintain consistent feature vector length, eigenvalues of the Laplacian cannot be directly concatenated due to varying complex sizes throughout the filtration. To address this, a length parameter is set to truncate or zero pad eigenvalues, ensuring uniform length regardless of the complex. Flattening eigenvalues across complex pairs and dimensions yields the final feature vector. With resolution R, max dimension D and truncating/zero padding eigenvalues to length L, we end with a $\frac{1}{2}R(R+1)DL$ dimensional feature vector. The entire process is captured in Figure 4, which showcases the pipeline.

5. Experimental Details

We evaluate the persistent Laplacian against other topological and geometric feature embedding techniques. The main baseline we are comparing against is persistent homology, as this is the primary technique used for topological embeddings in TDA. Our specific embedding of the persistence diagram (a summary of persistent homology over a filtration) once computed depends on the application: we typically use persistence images (Adams et al., 2017) but use other embeddings if they are better according to published baselines. We also compare against the graph Laplacian and (where there is data with more than edgewise interactions) the combinatorial Laplacian, as these represent the non-persistent versions of the k-persistent Laplacian for $k = 0$

Table 1: Results across Persistent Homology (PH), Graph Laplacian (GL), Combinatorial Laplacian (CL) and Persistent Laplacian (PL) methods, on the MNIST and MoleculeNet QM7/QM7b datasets. We report the accuracy for MNIST and the mean absolute error (MAE) for QM7/QM7b.

Dataset	PH	GL	CL	PL
MNIST (Accuracy)	0.625 ± 0.010	0.768 ± 0.006	0.798 ± 0.006	**0.821 ± 0.011**
QM7-3D (MAE)	92.43 ± 0.629	**69.23 ± 0.388**	**68.85 ± 0.421**	**68.86 ± 0.431**
QM7-CM Filtration 1 (MAE)	232.0 ± 6.435	**12.42 ± 0.191**	–	**12.38 ± 0.188**
QM7-CM Filtration 2 (MAE)	45.47 ± 1.438	23.19 ± 0.809	23.33 ± 0.785	**21.94 ± 0.737**
QM7b-CM-0 Filtration 1(MAE)	73.12 ± 1.893	**12.39 ± 0.599**	–	**12.35 ± 0.597**
QM7b-CM-0 Filtration 2 (MAE)	84.53 ± 1.804	**22.59 ± 0.535**	**22.85 ± 0.511**	26.14 ± 0.479

and $k > 0$ respectively. We run experiments on the MNIST handwritten digit dataset (LeCun et al., 2010) to provide a base demonstration of value in a machine learning setting, and the MoleculeNet dataset (Wu et al., 2018), a collection of molecular data along with chemical properties to predict. The MoleculeNet data serves to evaluate the applicability of the persistent Laplacian to realistic application scenarios. In the remainder of this section we expand on our exact methodology for each dataset and baseline, and specify precisely the data and tasks we consider for MoleculeNet.

MNIST is a standard dataset consisting of 28x28 grayscale images (LeCun et al., 2010). Each image is one handwritten digit, and the task is to classify that digit. Each of our baselines takes a filtration as input, so we need to embed grayscale images. According to the Giotto-TDA documentation[2] the best performing filtration for persistent homology is the height filtration with direction $[1, 0]$. To implement this we first threshold the image, discarding all pixels with a grayscale value less than 0.4 (which is in fact the optimal value for our primary competitor, persistent homology). The distance from the plane defined by our direction (in this case, the leftmost edge of the image) gives the value at which each cube (a pixel or 2x2 collection of pixels, as we are using a cubical complex) enters the filtration. We then compute the persistent Laplacian feature vector as defined in Section 4, with resolution $R = 4$ and using dimensions $p = 0, 1$. We compute the persisent homology and diagrams using Giotto-TDA, then embed it as they do, using persistence entropy and diagram amplitudes with a collection of metrics.[3] The graph Laplacian and combinatorial Laplacian are vectorized identically to the persistent Laplacian, only as they cannot work with persistence we simply compute them at the start values of the complex pairs. In the notation of Section 4, given discrete values of our filtration $F = [\alpha, \beta, , \dots]$, we compute $\Delta_p^{K_\alpha}, \Delta_p^{K_\beta}, \dots$ with $p = 0$ for the graph Laplacian and $p = 0, 1$ for the combinatorial Laplacian. We then embed the eigenvalues in the same way as the combinatorial Laplacian.

MoleculeNet is a collection of benchmark datasets designed to evaluate machine learning methods for prediction of molecular properties (Wu et al., 2018). We focus on two subsets of MoleculeNet, QM7 (Blum & Reymond, 2009; Rupp et al., 2012b) and QM7b (Blum & Reymond, 2009; Montavon et al., 2013). The data comes in two forms: 3D coordinates of the atoms in each molecule, and the Coulomb matrix of each molecule, a matrix $M = [m_{i,j}]_{i \in I}$ for some list of nuclei I, where $m_{i,j}$ is the electrostatis interaction between atomic nuclei in the molecule (Rupp et al., 2012b). We embed the 3D coordinates into a simplicial complex using the Vietoris-Rips construction (Vietoris, 1927): we add k points to the complex as a $(k-1)$-simplex when they are pairwise within distance ϵ of each other. The parameter ϵ then defines our filtration.

For the Coulomb matrices, we consider two types of filtrations. For the first filtration (Filtration 1), we let each nuclei $i \in I$ enter the filtration as a point at time 0, then adding edges ij at their electrostatic interaction value $m_{i,j}$. In this way we use the electrostatic interactions to induce the filtration. However, we note that for this particular filtration, the persistent Laplacian method will be almost identical to the graph Laplacian method. Recall from Section 2.5 that $\Delta_0^{K,L} = \Delta_0^L$ when K and L have the same vertex set. Hence, under our pipeline, for the filtration indexed by $\{K_i\}_{i=1}^N$, the graph Laplacian method extracts features from $\{\Delta_0^{K_1}, \Delta_0^{K_1}, \dots, \Delta_0^{K_N}\}$ whereas the persistent Laplacian uses features from $\{\Delta_0^{K_2}, \Delta_0^{K_3}, \dots, \Delta_0^{K_N}\}$ (we do not use 1-dim persistent Laplacians in this case since there will be no meaningful 1-dim up persistent Laplacians for graphs); note the subtle difference at starting point. Due to this similarity, to fully utilize the power of persistent Laplacian, we also consider another more complicated filtration (Filtration 2). In Filtration 2, a nuclei $i \in I$ is added into the filtration at $\min_{j \sim i} m_{ij}$ and a k-simplex $\sigma = \{i_0, \dots, i_k\}$ is added at $\max_{a,b} m_{i_a, i_b}$. For this filtration, we will use both 0 and 1

[2] `https://giotto-ai.github.io/gtda-docs/0.3.1/notebooks/MNIST_classification.html`

[3] Note that they also use 17 filtrations to produce their feature vector. We only use one filtration for all of our baselines, so we do the same in this experiment to make the information equal across the board. We select the directional filtration with direction $[1, 0]$, as that is the one they state has the best performance with TDA.

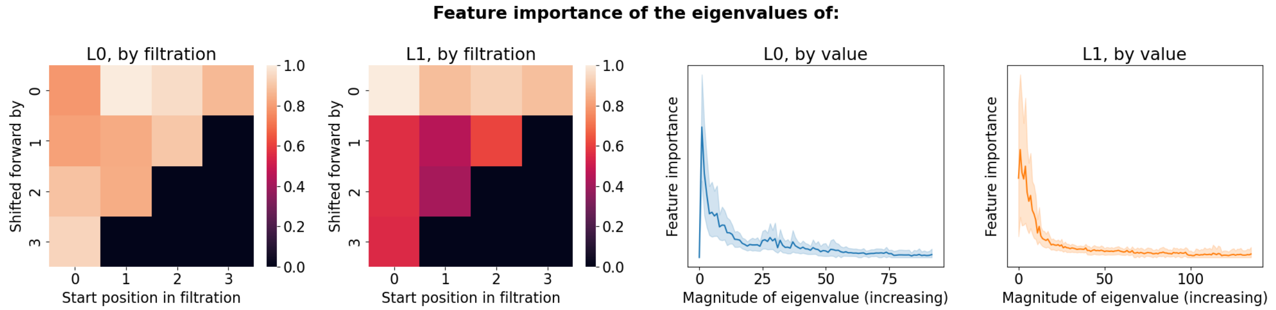


Figure 4: We aggregate feature importance by the complex pair and magnitude of the eigenvalues, and across dimensions 0 and 1. We can see that the persistence is being utilized, as there are discriminative features across the eigenvalues of each complex pair.

dimensional persistent Laplacians.

Each of the MoleculeNet tasks we consider is a regression task for which we must predict specific electrochemical properties. For QM7 the task is to predict the atomization energy of each molecule (Blum & Reymond, 2009), and we do so with both the 3D coordinates and Couloumb matrices (CM). For QM7b, there are only CM matrices available, and the specific tasks are reported in Appendix B, Table 2.

6. Results and Discussion

Table 1 shows our results for the persistent homology-based embeddings (PH), Graph Laplacian (GL), combinatorial Laplacian (CL, when higher-order interactions are present in the data), and the persistent Laplacian embeddings (PL). The reported score is the accuracy for the MNIST classification task, and the mean absolute error (MAE) for the remaining regression tasks.

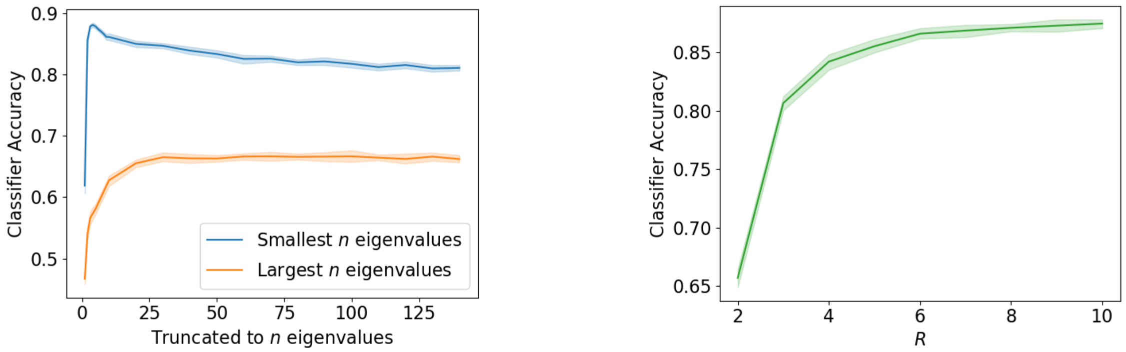
In this article, we aim to demonstrate the value of the persistent Laplacian at incorporating and extending persistent homology, the primary tool in TDA. Looking at Table 1, we see that we typically exceed the performance of persistent homology with the persistent Laplacian embedding across both MNIST and MoleculeNet tasks. Particularly interesting is the performance of the persistent Laplacian on MNIST, attaining 82% performance with a simple embedding and vastly outperforming the 62.5% accuracy attained by persistent homology. We consider this performance strong evidence for the suitability of the persistent Laplacian for feature embeddings in Topological Data Analysis.

For the MoleculeNet QM7 and QM7b datasets (for space concern, we only included one task for QM7b dataset in Table 1; see full results in Table 3), the persistent Laplacian also outperforms persistent homology in most cases, although there are some tasks on which PH performs better.

The fact that the persistent Laplacian outperforms the graph and combinatorial Laplacian on MNIST, suggests that the filtration contains vital information in this case. This makes sense, as in our choice of filtration the digit is ‘unveiled’ over the course of the filtration. On QM7 and QM7b, on the other hand, we see an almost identical performance between the persistent Laplacian and its non-persistent versions for Filtration 1 as we expected since the persistent Laplacian method is almost identical to the graph Laplacian method for Filtration 1 regarding the Coulomb matrices. For Filtration 2, however, although PL features should theoretically contain almost all GL features, we do not see much improvement in performance (sometimes the performances are worse); this may be due to the fact that our choice of filtration does not encode enough additional information in this dataset. However, we remark that using graph Laplacians across filtrations is not standard in the literature (this has been explored in only a handful of studies, which we cover in the Related Work, Section 1.1) which should be viewed as a simple instance of our persistent Laplacian method.

6.1. Feature Importance

We also evaluate the feature importance that the models assign to each eigenvalue, in terms of the Mean Decrease in Impurity metric (Scornet, 2023). In Figure 4, we display the feature importance grouped both by the complex pair (i.e., the position in the filtration), and by the size of the eigenvalue, in order to ascertain the effect of that on the discriminative information it contains. For the complex pair feature importance plots,our x axis is the start value of the filtration, and the y axis is the shift forwards through the filtration. For example, if the start value is 1 and the shift forward is 2, then our complex pair is $K_1 \hookrightarrow K_3$. We compute the feature importance plots for the MNIST model. From the feature importance by filtration plots (Figure 4, left), it is clear that the model is using features from each of the complex pairs that we generate. This means that *persistence is providing valuable discriminative information* that the model is utilizing. This is particularly true in the



(a) The effect of truncation.

(b) The effect of R.

Figure 5: We investigate the effect of eigenvalue truncation and the parameter R on the efficacy of the persistent Laplacian on the MNIST classification task.

0-persistent Laplacian, but also for the 1-persistent Laplacian. Next, we consider the plots that relate the size of the eigenvalue to the feature importance (Figure 4, right). Note that the smallest eigenvalue of the 0-persistent Laplacian has zero importance. This makes sense, as we know from Section 2 that this eigenvalue will always be 0, as there is always at least one connected component, and the nullity of 0-PL is the number of connected components. The spike immediately following it also makes sense, as the number of following zero eigenvalues gives topological information, namely the number of connected components, and the smallest following non-zero eigenvalues are known to contain significant information about the structure of the graph. Moving onto the 1-persistent Laplacian, we see, unsurprisingly, that the zero eigenvalues having high importance, as they quantify the number of holes present in the complex. It is interesting that there remains high feature importance for the small non-zero eigenvalues, as this implies they are also capturing discriminative information.

6.2. Ablation Studies

Embedding the persistent Laplacian into a feature vector over a whole filtration as we describe in Section 4 requires two parameters: the truncation parameter, which determines how many of the eigenvalues in the spectra of each persistent Laplacian we consider, and the resolution R, which determines how we uniformly sample the continuous filtration to choose complex pairs to compute the persistent Laplacian of. We ran ablation studies using the MNIST dataset to consider how each of these choices effects the performance of the downstream classification task. The results of these experiments are shown in Figure 5.

Truncation. We investigated two different truncation strategies: truncating to the smallest n eigenvalues and to the largest n eigenvalues, the results of which are shown in Figure 5a. We found that truncating to the smallest eigenvalues (i.e., dropping the large eigenvalues) performed significantly better than truncating the largest eigenvalues (i.e., dropping the small eigenvalues). In practice, then, we would recommend using only a small number of eigenvalues - the exact number will be data-specific.

Resolution. For very low values of R, the performance is not good - presumably because the model cannot exploit the additional information contained in the filtration. As we increase R, and finer details about the filtration become available via the feature vector, performance increases.

6.3. Non-Topological Baselines

We also compare our results to non-topological baselines. In particular, for MNIST we evaluate a Random Forest on the flattened raw images and a CNN implemented in PyTorch. For MoleculeNet, we evaluate a Random Forest and Kernel Ridge Regression, which is in the literature as the best performing ‘traditional’ technique. We also report Deep Tensor Neural Networks (Schütt et al., 2017) and Gated Graph Recurrent Neural Networks Shindo & Matsumoto (2019) results from the literature. Typically the persistent Laplacian outperforms shallow methods but is worse than deep methods - we discuss this in detail in Appendix C.

7. Conclusions

We have demonstrated that the persistent Laplacian can consistently outperform the widely used persistent homology by incorporating non-topological information in the filtration representation of a data set. This suggests that this new theoretical development, which is able to seamlessly and efficiently combine topological and geometric information, should be more broadly researched and utilized in data analysis applications.

References

Adams, H., Emerson, T., Kirby, M., Neville, R., Peterson, C., Shipman, P., Chepushtanova, S., Hanson, E., Motta, F., and Ziegelmeier, L. Persistence images: A stable vector representation of persistent homology. *Journal of Machine Learning Research*, 18(8):1–35, 2017. URL http://jmlr.org/papers/v18/16-337.html.

Belle, V. and Papantonis, I. Principles and practice of explainable machine learning. *Frontiers in Big Data*, 4, July 2021. doi: 10.3389/fdata.2021.688969. URL https://doi.org/10.3389/fdata.2021.688969.

Blum, L. C. and Reymond, J.-L. 970 million druglike small molecules for virtual screening in the chemical universe database GDB-13. *J. Am. Chem. Soc.*, 131:8732, 2009.

Bodnar, C., Frasca, F., Wang, Y., Otter, N., Montufar, G. F., Lio, P., and Bronstein, M. Weisfeiler and lehman go topological: Message passing simplicial networks. In *International Conference on Machine Learning*, pp. 1026–1037. PMLR, 2021.

Bukkuri, A., Andor, N., and Darcy, I. K. Applications of topological data analysis in oncology. *Frontiers in Artificial Intelligence*, 4, 2021. ISSN 2624-8212. doi: 10.3389/frai.2021.659037. URL https://www.frontiersin.org/articles/10.3389/frai.2021.659037.

Dey, T. K. and Wang, Y. *Computational topology for data analysis*. Cambridge University Press, 2022.

Duval, A. M., Klivans, C. J., and Martin, J. L. Cellular spanning trees and laplacians of cubical complexes. *Advances in Applied Mathematics*, 46(1-4):247–274, 2011.

Eckmann, B. Harmonische funktionen und randwertaufgaben in einem komplex. *Commentarii Mathematici Helvetici*, 17(1):240–255, 1944.

Edelsbrunner, Letscher, and Zomorodian. Topological persistence and simplification. *Discrete & computational geometry*, 28:511–533, 2002.

Edelsbrunner, H. and Harer, J. L. *Computational topology: an introduction*. American Mathematical Society, 2022.

Garin, A. and Tauzin, G. A topological "reading" lesson: Classification of MNIST using TDA. *CoRR*, abs/1910.08345, 2019. URL http://arxiv.org/abs/1910.08345.

Goldberg, T. E. Combinatorial laplacians of simplicial complexes. *Senior Thesis, Bard College*, 2002.

Gong, W., Wee, J., Wu, M.-C., Sun, X., Li, C., and Xia, K. Persistent spectral simplicial complex-based machine learning for chromosomal structural analysis in cellular differentiation. *Briefings in Bioinformatics*, 23(4): bbac168, 2022.

Gülen, A. B., Mémoli, F., Wan, Z., and Wang, Y. A Generalization of the Persistent Laplacian to Simplicial Maps. In *39th International Symposium on Computational Geometry (SoCG 2023)*, 2023.

Hensel, F., Moor, M., and Rieck, B. A survey of topological machine learning methods. *Frontiers in Artificial Intelligence*, 4, May 2021. doi: 10.3389/frai.2021.681108. URL https://doi.org/10.3389/frai.2021.681108.

Horak, D. and Jost, J. Spectra of combinatorial laplace operators on simplicial complexes. *Advances in Mathematics*, 244:303–336, 2013.

Jean-Paul, S., Elseify, T., Obeid, I., and Picone, J. Issues in the reproducibility of deep learning results. In *2019 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, pp. 1–4, 2019. doi: 10.1109/SPMB47826.2019.9037840.

Jiang, P., Chi, Y., Li, X.-S., Liu, X., Hua, X.-S., and Xia, K. Molecular persistent spectral image (mol-psi) representation for machine learning models in drug design. *Briefings in Bioinformatics*, 23(1):bbab527, 2022.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.

LeCun, Y. and Bengio, Y. *Convolutional Networks for Images, Speech, and Time Series*, pp. 255–258. MIT Press, Cambridge, MA, USA, 1998. ISBN 0262511029.

LeCun, Y., Cortes, C., and Burges, C. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

Lieutier, A. Talk: Persistent harmonic forms, 2014. URL https://project.inria.fr/gudhi/files/2014/10/Persistent-Harmonic-Forms.pdf.

Mémoli, F., Wan, Z., and Wang, Y. Persistent laplacians: Properties, algorithms and implications. *SIAM Journal on Mathematics of Data Science*, 4(2):858–884, 2022.

Meng, Z. and Xia, K. Persistent spectral&#x2013;based machine learning (perspect ml) for protein-ligand binding affinity prediction. *Science Advances*, 7(19):eabc5329, 2021. doi: 10.1126/sciadv.abc5329.

Montavon, G., Rupp, M., Gobre, V., Vazquez-Mayagoitia, A., Hansen, K., Tkatchenko, A., Müller, K.-R., and von Lilienfeld, O. A. Machine learning of molecular electronic properties in chemical compound space. *New Journal of Physics*, 15(9): 095003, 2013. URL `http://stacks.iop.org/1367-2630/15/i=9/a=095003`.

Pritchard, Y., Sharma, A., Clarkin, C., Ogden, H., Mahajan, S., and Sánchez-García, R. J. Persistent homology analysis distinguishes pathological bone microstructure in non-linear microscopy images. *Scientific Reports*, 2022.

Qiu, Y. and Wei, G.-W. Persistent spectral theory-guided protein engineering. *Nature Computational Science*, 3(2): 149–163, 2023.

Rupp, M., Tkatchenko, A., Müller, K.-R., and von Lilienfeld, O. A. Fast and accurate modeling of molecular atomization energies with machine learning. *Phys. Rev. Lett.*, 108:058301, Jan 2012a. doi: 10.1103/PhysRevLett.108.058301. URL `https://link.aps.org/doi/10.1103/PhysRevLett.108.058301`.

Rupp, M., Tkatchenko, A., Müller, K.-R., and von Lilienfeld, O. A. Fast and accurate modeling of molecular atomization energies with machine learning. *Physical Review Letters*, 108:058301, 2012b.

Schütt, K. T., Arbabzadah, F., Chmiela, S., Müller, K. R., and Tkatchenko, A. Quantum-chemical insights from deep tensor neural networks. *Nature Communications*, 8(1), January 2017. doi: 10.1038/ncomms13890. URL `https://doi.org/10.1038/ncomms13890`.

Scornet, E. Trees, forests, and impurity-based variable importance in regression. In *Annales de l'Institut Henri Poincare (B) Probabilites et statistiques*, volume 59, pp. 21–52. Institut Henri Poincaré, 2023.

Shindo, H. and Matsumoto, Y. Gated graph recursive neural networks for molecular property prediction, 2019. URL `https://arxiv.org/abs/1909.00259`.

Strömbom, D. Persistent homology in the cubical setting: theory, implementations and applications, 2007.

Tauzin, G., Lupo, U., Tunstall, L., Pérez, J. B., Caorsi, M., Medina-Mardones, A., Dassatti, A., and Hess, K. giotto-tda: A topological data analysis toolkit for machine learning and data exploration, 2020.

Vietoris, L. Über den höheren zusammenhang kompakter räume und eine klasse von zusammenhangstreuen abbildungen. *Mathematische Annalen*, 97(1):454–472, 1927.

Von Luxburg, U. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

Wang, R., Nguyen, D. D., and Wei, G.-W. Persistent spectral graph. *International journal for numerical methods in biomedical engineering*, 36(9):e3376, 2020.

Wang, R., Zhao, R., Ribando-Gros, E., Chen, J., Tong, Y., and Wei, G.-W. Hermes: Persistent spectral graph software. *Foundations of data science (Springfield, Mo.)*, 3(1):67, 2021.

Wee, J. and Xia, K. Persistent spectral based ensemble learning (perspect-el) for protein–protein binding affinity prediction. *Briefings in Bioinformatics*, 23(2), 2022.

Wu, Z., Ramsundar, B., Feinberg, E. N., Gomes, J., Geniesse, C., Pappu, A. S., Leswing, K., and Pande, V. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

A. The Schur Complement Formulation for the Persistent Laplacian

In Gülen et al. (2023), the Schur complement is interpreted as an operator, as follows. Let M denote the matrix representation of some operator $\mathcal{M} : \mathbb{R}^n \to \mathbb{R}^n$ and let V denote the vector space spanned by the first $n - d$ columns of M. We explicitly write $M = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \in \mathbb{R}^{n \times n}$ as a block matrix where D is a $d \times d$ matrix. Then, M/D is the matrix representation of an operator called the *Schur restriction* of $\mathcal{M}$ on V, denoted by $\mathbf{Sch}(\mathcal{M}, V)$, with respect to the already chosen basis on V (i.e., the first $n - d$ columns of M).

Lemma A.1 ((Gülen et al., 2023, Proposition 3.2)). *Let $f : U \to V$ be a linear map between finite dimensional linear product spaces. Let $W \subseteq V$ be a subspace and let $f_W := f|_{f^{-1}(W)}$ be the restriction of f on the linear subspace $f^{-1}(W)$ of U. Consider the operator $\mathcal{L} := f \circ f^* : V \to V$. Then,*

$$\mathbf{Sch}(\mathcal{L}, W) = f_W \circ f_W^*.$$

Now, we let $U = C_{p+1}^L$, $V = C_p^L$, $f = \partial_{p+1}^L$ and $W = C_p^K$. It is easy to check that $\mathcal{L} = f \circ f^* = \Delta_{\mathrm{up},p}^L$, $f^{-1}(W) = C_{p+1}^{L,K}$ and $f_W = \partial_{p+1}^{L,K}$. Hence, by definition one has $f_W \circ f_W^* = \Delta_{\mathrm{up},p}^{K,L}$. Now, since $\mathbf{\Delta}_{\mathrm{up},p}^L / \mathbf{\Delta}_{\mathrm{up},p}^L(I_K^L, I_K^L)$ is the matrix representation of $\mathbf{Sch}(\mathcal{L}, W)$, by applying the above lemma one has that $\mathbf{\Delta}_{\mathrm{up},p}^L / \mathbf{\Delta}_{\mathrm{up},p}^L(I_K^L, I_K^L)$ is the matrix representation of the up persistent Laplacian $\Delta_{\mathrm{up},p}^{K,L}$.

B. Additional Experimental Details

The specifics of the MoleculeNet QM7b tasks that we evaluate are in Table 2. Full results regarding tasks for QM7 and QM7b datasets with Coulomb matrices are provided in Table 3 (for Filtration 1) and Table 4 (for Filtration 2).

Table 2: The QM7b tasks are as follows. Each of these is a property of the molecule being described, evaluated with different methods. For full details of the tasks please see Montavon et al. (2013).

Task ID	Evaluation Method	Description
0	PBE0	Activation energy
1	ZINDO	Excitation energy with the most absorption
2	ZINDO	Highest absorption
3	ZINDO	HOMO
4	ZINDO	LUMO
5	ZINDO	1st excitation energy
6	ZINDO	Ionization potential
7	ZINDO	Electron affinity
8	KS	HOMO
9	KS	LUMO
10	GW	HOMO
11	GW	LUMO
12	PBE	Polarisability
13	SCS	Polarisability

C. Comparison to Non-topological Methods

We have demonstrated that the persistent Laplacian can outperform other topological baselines in MNIST and consistently outperform persistent homology in MoleculeNet, as well as discussed the additional information it can represent due to its theoretical properties. We consider some additional baselines here. Firstly, for MNIST, we evaluated the efficacy of using the flattened raw image as input into a random forest, as well as a convolutional neural network (LeCun & Bengio, 1998). Both significantly outperformed our topological methods, with the random forest scoring an accuracy of 0.9387 ± 0.0022 and the CNN scoring an accuracy of 0.9918 ± 0.0002. In comparison, the persistent Laplacian is the best scoring topological method, with an accuracy of 0.821 ± 0.011 (with $R = 5$ and truncating to the mean number of eigenvalues). In Table 5 we compare the performance of the persistent Laplacian to techniques from shallow and deep learning. Note that we do not use the QM7-3D dataset, as the coulomb matrix is typically used as the representation of the data for non-topological methods Rupp et al. (2012a). In particular, we evaluated the performance of random forests (RF) and kernel ridge regression (KRR)

Table 3: Results across Persistent Homology (PH), Graph Laplacian (GL), Combinatorial Laplacian (CL) and Persistent Laplacian (PL) methods, on all QM7b-CM tasks, using Filtration 1. We report the mean absolute error (MAE) for QM7/QM7b.

Dataset	PH	GL	CL	PL
QM7-CM	232.0 ± 6.435	12.42 ± 0.191	–	12.38 ± 0.188
QM7b-CM-0	73.12 ± 1.893	12.39 ± 0.599	–	12.35 ± 0.597
QM7b-CM-1	1.916 ± 0.047	1.731 ± 0.038	–	1.729 ± 0.037
QM7b-CM-2	0.122 ± 0.004	0.096 ± 0.004	–	0.095 ± 0.004
QM7b-CM-3	0.484 ± 0.009	0.393 ± 0.008	–	0.393 ± 0.007
QM7b-CM-4	0.510 ± 0.009	0.362 ± 0.008	–	0.361 ± 0.007
QM7b-CM-5	0.651 ± 0.027	0.427 ± 0.010	–	0.426 ± 0.011
QM7b-CM-6	0.485 ± 0.011	0.406 ± 0.006	–	0.406 ± 0.007
QM7b-CM-7	0.548 ± 0.009	0.393 ± 0.010	–	0.392 ± 0.009
QM7b-CM-8	0.302 ± 0.005	0.272 ± 0.005	–	0.272 ± 0.005
QM7b-CM-9	0.305 ± 0.009	0.226 ± 0.006	–	0.226 ± 0.007
QM7b-CM-10	0.335 ± 0.007	0.306 ± 0.006	–	0.305 ± 0.006
QM7b-CM-11	0.236 ± 0.006	0.197 ± 0.004	–	0.197 ± 0.004
QM7b-CM-12	0.696 ± 0.015	0.352 ± 0.011	–	0.352 ± 0.012
QM7b-CM-13	0.726 ± 0.016	0.285 ± 0.009	–	0.284 ± 0.009

Table 4: Results across Persistent Homology (PH), Graph Laplacian (GL), Combinatorial Laplacian (CL) and Persistent Laplacian (PL) methods, on all QM7b-CM tasks, using Filtration 2. We report the mean absolute error (MAE) for QM7/QM7b.

Dataset	PH	GL	CL	PL
QM7-CM	45.47 ± 1.438	23.19 ± 0.809	23.33 ± 0.785	21.94 ± 0.737
QM7b-CM-0	84.53 ± 1.804	22.59 ± 0.535	22.85 ± 0.511	26.14 ± 0.479
QM7b-CM-1	2.585 ± 0.033	1.988 ± 0.046	2.014 ± 0.048	2.236 ± 0.035
QM7b-CM-2	0.151 ± 0.004	0.111 ± 0.005	0.112 ± 0.005	0.122 ± 0.005
QM7b-CM-3	0.785 ± 0.013	0.517 ± 0.013	0.521 ± 0.011	0.586 ± 0.012
QM7b-CM-4	0.884 ± 0.020	0.481 ± 0.017	0.481 ± 0.017	0.526 ± 0.012
QM7b-CM-5	1.219 ± 0.044	0.612 ± 0.011	0.611 ± 0.014	0.683 ± 0.014
QM7b-CM-6	0.790 ± 0.012	0.533 ± 0.015	0.536 ± 0.014	0.600 ± 0.013
QM7b-CM-7	0.973 ± 0.025	0.532 ± 0.020	0.532 ± 0.019	0.586 ± 0.014
QM7b-CM-8	0.496 ± 0.009	0.346 ± 0.009	0.345 ± 0.008	0.385 ± 0.008
QM7b-CM-9	0.431 ± 0.012	0.263 ± 0.008	0.263 ± 0.009	0.283 ± 0.008
QM7b-CM-10	0.558 ± 0.007	0.393 ± 0.010	0.394 ± 0.010	0.436 ± 0.008
QM7b-CM-11	0.300 ± 0.009	0.225 ± 0.007	0.226 ± 0.007	0.239 ± 0.005
QM7b-CM-12	0.883 ± 0.028	0.458 ± 0.013	0.462 ± 0.013	0.545 ± 0.011
QM7b-CM-13	0.866 ± 0.029	0.376 ± 0.008	0.384 ± 0.008	0.465 ± 0.010

on the flattened coulomb matrices. We also report the deep learning SOTA using deep tensor neural networks (DTNN) (Wu et al., 2018; Schütt et al., 2017) and gated graph recurrent neural networks (Shindo & Matsumoto, 2019). We find that we outperform the shallow methods, but are once again beaten by methods from deep learning.

We would argue that despite methods from deep learning beating the persistent Laplacian as a feature vector, we retain several advantages over deep methods.

(i) Firstly, the persistent Laplacian is being used as a feature vectorization for downstream input into a simple classifier. It is a technique that relies heavily on theory, making the feature vector itself related to real-world understanding of shape and structure. In comparison, deep learning is famously a black box, and attempts to improve explainability and interpretability struggle in practice Belle & Papantonis (2021).

(ii) Secondly, the featurization of the persistent Laplacian relies only on two understandable parameters (Section 6.2), and is easily reproducible. In comparison, deep learning techniques are often very sensitive to many opaque hyperparameters. Reproducing reported models is often impossible, with the ‘reproducibility crisis’ a known problem in the deep learning research community (Jean-Paul et al., 2019) .

Despite these points, clearly deep learning is a vastly powerful tool. In fact, topological tools are often at their most powerful when partnered with deep learning, as evidenced by the increasing popularity of Topological Machine Learning (Hensel et al., 2021). Research such as ours, which introduces a featurization of the persistent Laplacian for embedding data and evaluates the persistent Laplacian on ML baselines, provides a strong base for future work on the integration of the persistent Laplacian into Topological machine learning.

Table 5: We also compare the persistent Laplacian to non-topological baselines. We trained random forests (RF) and kernel ridge regression (KRR) models on the flattened Coulomb matrices, as is standard in the literature (Rupp et al., 2012a). We also report the results from the SOTA in deep learning: deep tensor neural networks (DTNN) (Schütt et al., 2017) and gated graph recurrent neural networks (GGRNN) (Shindo & Matsumoto, 2019). Our techniques outperform the shallow methods, but are beaten by deep learning. In the discussion we consider the place of topological techniques within the context of deep learning.

	Shallow Learning		TDA		Deep Learning	
Dataset	RF	KRR	PL (Filt. 1)	PL (Filt. 2)	DTNN	GGRNN
QM7-CM	11.20 ± 0.261	26.16 ± 0.651	12.38 ± 0.188	21.936 ± 0.737	8.75	–
QM7b-CM-0	38.17 ± 0.730	140.0 ± 3.752	12.35 ± 0.597	26.141 ± 0.479	21.5	13.7
QM7b-CM-1	2.473 ± 0.027	2.668 ± 0.061	1.729 ± 0.037	2.236 ± 0.035	1.26	1.02
QM7b-CM-2	0.130 ± 0.003	0.164 ± 0.004	0.095 ± 0.004	0.122 ± 0.005	0.074	0.072
QM7b-CM-3	0.708 ± 0.011	1.192 ± 0.042	0.393 ± 0.007	0.586 ± 0.012	0.192	0.140
QM7b-CM-4	0.768 ± 0.016	0.813 ± 0.02	0.361 ± 0.007	0.526 ± 0.012	0.159	0.0915
QM7b-CM-5	1.057 ± 0.022	1.185 ± 0.034	0.426 ± 0.011	0.683 ± 0.014	0.296	0.121
QM7b-CM-6	0.713 ± 0.011	1.181 ± 0.043	0.406 ± 0.007	0.600 ± 0.013	0.214	0.176
QM7b-CM-7	0.853 ± 0.016	0.896 ± 0.022	0.392 ± 0.009	0.586 ± 0.014	0.174	0.0940
QM7b-CM-8	0.437 ± 0.009	0.906 ± 0.027	0.272 ± 0.005	0.385 ± 0.008	0.155	0.142
QM7b-CM-9	0.376 ± 0.012	0.426 ± 0.013	0.226 ± 0.007	0.283 ± 0.008	0.129	0.092
QM7b-CM-10	0.495 ± 0.010	1.134 ± 0.036	0.305 ± 0.006	0.436 ± 0.008	0.166	0.142
QM7b-CM-11	0.271 ± 0.007	0.312 ± 0.007	0.197 ± 0.004	0.239 ± 0.005	0.139	0.118
QM7b-CM-12	0.693 ± 0.017	1.348 ± 0.026	0.352 ± 0.012	0.545 ± 0.011	0.173	0.100
QM7b-CM-13	0.624 ± 0.015	1.362 ± 0.029	0.284 ± 0.009	0.465 ± 0.010	0.149	0.0578

D. Runtimes

We report the runtimes for pre-processing (embedding data into filtrations of simplicial/cubical complexes), embeddings (computing the topological summaries), and post-processing (mapping the topological summaries into a persistence diagram). Note that each of these timing experiments are run on one core of a 2021 MacBook Pro with an M1 Pro chip. We report the time in seconds. Note that in these tables we reference others' implementation; if there is no reference then it is our implementation. We repeat each timing experiment 100 times. The error bounds are typically in the order of 10^{-5}, so for readability we do not report them. Note that the time taken for the computation of the persistent Laplacian for one simplicial complex pair is of the same order of magnitude as the persistent Laplacian. As you increase R, and thus the number of persistent Laplacians computed increases, the time taken will similarly increase. The time taken for the cubical complex persistent Laplacian is several orders of magnitude larger - this is due to our implementation of the cubical boundary.

Table 6: Time taken to pre-process the raw data, in seconds.

Pre-Processing	Time (seconds)
Height Filtration[4]	0.000376
Induced from coulomb matrix (QM7/QM7b)	0.000376

Table 7: Time taken to embed the pre-processed data, in seconds.

Embedding	Time (seconds)
Persistent Homology ($p = 0, 1$)[5]	0.000231
Cubical Homology ($p = 0, 1$)[4]	0.000231
Graph Laplacian	0.0000292
Combinatorial Laplacian	0.000382
Persistent Laplacian ($p = 0, 1$, simplicial complex pair)	0.000380
Persistent Laplacian ($p = 0, 1$, cubical complex pair)	1.855

Table 8: Time taken to post-process the embedded data, in seconds.

Post-Processing	Time (seconds)
MNIST Baseline Embedding[4] (Garin & Tauzin, 2019)	0.000376
Persistence Images[6]	0.00188
Persistent Laplacian Vectorization (Section 4)	0.00330

Table 9: Time taken to compute Vietoris-Rips persistence, in seconds.

Post-Processing	Time (seconds)
Vietoris-Rips Persistence ($p = 0, 1$) [4]	0.000578
Vietoris-Rips Persistence ($p = 0, 1, 2$) [4]	0.000912

As we used the Vietoris-Rips persistence function in Giotto-TDA (Tauzin et al., 2020) for the QM7 3D coordinate experiments we cannot disentangle the time taken for pre-processing and computing the topological embedding. We report the total time to compute the Vietoris-Rips persistent homology of one set of 3d coordinates from QM7.

[4] Using Giotto-TDA (Tauzin et al., 2020).

[5] Using Dionysus 2.

[6] Using Persim.