



# Optimization of Spectral Wavelets for Persistence-Based Graph Classification

Ka Man Yim\* and Jacob Leygonie

Mathematical Institute, University of Oxford, Oxford, United Kingdom

A graph's spectral wavelet signature determines a filtration, and consequently an associated set of extended persistence diagrams. We propose a framework that optimizes the choice of wavelet for a dataset of graphs, such that their associated persistence diagrams capture features of the graphs that are best suited to a given data science problem. Since the spectral wavelet signature of a graph is derived from its Laplacian, our framework encodes geometric properties of graphs in their associated persistence diagrams and can be applied to graphs without a priori node attributes. We apply our framework to graph classification problems and obtain performances competitive with other persistence-based architectures. To provide the underlying theoretical foundations, we extend the differentiability result for ordinary persistent homology to extended persistent homology.

**Keywords:** topological data analysis, graph classification, graph Laplacian, extended persistent homology, spectral wavelet signatures, radial basis neural network

## OPEN ACCESS

### Edited by:

Umberto Lupo,  
École Polytechnique Fédérale de  
Lausanne, Switzerland

### Reviewed by:

Bastian Rieck,  
ETH Zürich, Switzerland  
Primož Skraba,  
Queen Mary University of London,  
United Kingdom

### \*Correspondence:

Ka Man Yim  
yim@maths.ox.ac.uk

### Specialty section:

This article was submitted to  
Mathematics of Computation and  
Data Science,  
a section of the journal  
Frontiers in Applied Mathematics and  
Statistics

**Received:** 09 January 2021

**Accepted:** 24 February 2021

**Published:** 22 April 2021

### Citation:

Yim KM and Leygonie J (2021)  
Optimization of Spectral Wavelets for  
Persistence-Based Graph  
Classification.  
Front. Appl. Math. Stat. 7:651467.  
doi: 10.3389/fams.2021.651467

## 1. INTRODUCTION

### 1.1. Background

Graph classification is a challenging problem in machine learning. Unlike data represented in Euclidean space, there is no easily computable notion of distance or similarity between graphs. As such, graph classification requires techniques that lie beyond mainstream machine learning techniques focused on Euclidean data. Much research has been conducted on methods such as graph neural networks (GNNs) [1] and graph kernels [2, 3] that embed graphs in Euclidean space in a consistent manner.

Recently, *persistent homology* [4, 5] has been applied as a feature map that explicitly represents topological and geometric features of a graph as a set of *persistence diagrams* (a.k.a. *barcodes*). In the context of our discussion, the persistent homology of a graph  $G = (V, E)$  depends on a vertex function  $f : V \rightarrow \mathbb{R}$ . In the case where a vertex function is not given with the data, several schemes have been proposed in the literature to assign vertex functions to graphs in a consistent way. For example, vertex functions can be constructed using local geometric descriptions of vertex neighborhoods, such as discrete curvature [6], heat kernel signatures [7] and Weisfeiler–Lehman graph kernels [8].

However, it is often difficult to know *a priori* whether a heuristic vertex assignment scheme will perform well in addressing different data science problems. For a single graph, we can optimize the vertex function over  $|V|$  many degrees of freedom in  $\mathbb{R}^V$ . In recent years, there have been many other examples of persistence optimization in data science applications. The first two examples of persistence optimization are the computation of Fréchet mean of barcodes using gradients on Alexandrov spaces [9], and that of point cloud inference [10], where a point cloud is optimized so

that its barcode fits a target fixed barcode. The latter is an instance of topological inverse problems (see Oudot and Solomon [11] for a recent overview of such). Another inverse problem is that of surface reconstruction [12]. Besides, in the context of shape matching [13], persistence optimization is used in order to learn an adequate function between shapes. Finally, there are also many recent applications of persistence optimization in Machine Learning, such as the incorporation of topological information in Generative Modeling [14–16] or in Image Segmentation [17, 18], the design of topological losses for Regularization in supervised learning [19] or for dimension reduction [20].

Each of these applications can be thought of as minimizing a certain *loss* function over a manifold  $\mathcal{M}$  of parameters:

$$\min_{\theta \in \mathcal{M}} \mathcal{L}(\theta),$$

where  $\mathcal{L} : \mathcal{M} \rightarrow \mathbf{Bar}^N \rightarrow \mathbb{R}$  factors through the space  $\mathbf{Bar}^N$  of  $N$ -tuples of barcodes. The aim is to find the parameter  $\theta$  that best fits the application at hand. Gradient descent is a very popular approach in minimization, but it requires the ability to differentiate the loss function. In fact, Leygonie et al. [21] provide notions of differentiability for maps in and out  $\mathbf{Bar}$  that are compatible with smooth calculus, and show that the loss functions  $\mathcal{L}$  corresponding the applications cited in the above paragraph are generically differentiable. The use of (stochastic) gradient descent is further legitimated by Carriere et al. [22], where convergence guarantees on persistence optimization problems are devised, using a recent study of stratified non-smooth optimization problems [23]. In practice, the minimization of  $\mathcal{L}$  can be unstable due to its non-convexity and partial non-differentiability. Some research has been conducted in order to smooth and regularize the optimization procedure [24, 25].

In a supervised learning setting, we want to optimize our vertex function assignment scheme over many individual graphs in a dataset. Since graphs may not share the same vertex set and come in different sizes, optimizing over the  $|V|$  degrees of freedom of any one graph is not conducive to learning a vertex function assignment scheme that can generalize to another graph. The degrees of freedom in any practical vertex assignment scheme should be independent of the number of vertices of a graph. However, a framework for parameterizing and optimizing the vertex functions of many graphs over a common parameter space  $\mathcal{M}$  is not immediately apparent.

The first instance of a graph persistence optimization framework (GFL) [26] uses a one layer graph isomorphism network (GIN) [1] to parameterize vertex functions. The GIN learns a vertex function by exploiting the local topology around each vertex. In this paper, we propose a different framework for assigning and parameterizing vertex functions, based on a graph's Laplacian operator. Using the Laplacian, we can explicitly take both local and global structures of the graph into consideration in an interpretable and transparent manner.

### 1.2. Outline and Contributions

We address the issue of vertex function parameterization and optimization using *wavelet signatures*. Wavelet signatures are

vertex functions derived from the eigenvalues and eigenvectors of the graph Laplacian and encode multiscale geometric information about the graph [27]. The wavelet signature of a graph is dependent on a choice of wavelet  $g : \mathbb{R} \rightarrow \mathbb{R}$ , a function on the eigenvalues of the graph's Laplacian matrix. We can thus obtain a parameterization of vertex functions for any graph  $F : \mathcal{M} \rightarrow \mathbb{R}^V$  by parameterizing  $g$ . Consequently, the *extended persistence* of a graph—which has only four non-trivial persistence diagrams—can be varied over the parameter space  $\mathcal{M}$ . If we have a function  $\text{Out} : \mathbf{Bar}^4 \rightarrow \mathbb{R}$  on persistence diagrams that we wish to minimize, we can optimize over  $\mathcal{M}$  to minimize the loss function

$$\mathcal{L} : \mathcal{M} \xrightarrow{F} \mathbb{R}^V \xrightarrow{\text{EPH}} \mathbf{Bar}^4 \xrightarrow{\text{Out}} \mathbb{R}. \quad (1)$$

If  $\mathcal{L}$  is generically differentiable, we can optimize the wavelet signature parameters  $\theta \in \mathcal{M}$  using gradient descent methods. We illustrate an application of this framework to a graph classification problem in **Figure 1**, where the loss function  $\mathcal{L}$  is the classification error of a graph classification prediction model based on the graph's extended persistence diagrams.

In section 2, we describe the assignment of vertex functions  $F : \mathcal{M} \rightarrow \mathbb{R}^V$  by reviewing the definition of wavelet signatures. While spectral wavelets have been used in graph neural network architectures that predict vertex features [1] and compress vertex functions [28], they have not been considered in a persistent homology framework for graph classification. We describe several ways to parameterize wavelets. We also show in Proposition 2.2 that wavelet signatures are independent of the choice of eigenbasis of the graph Laplacian from which it is derived, ensuring that it is well-defined. We prove this result in Appendix B in **Supplementary Material**.

In section 3, we describe the theoretical basis for optimizing the *extended* persistent homology of a vertex function  $\text{EPH} : \mathbb{R}^V \rightarrow \mathbf{Bar}^4$  and elucidate what it means for  $\mathcal{L}$  to be differentiable. In Proposition 3.3, we generalize the differentiability formalism of ordinary persistence [21] to extended persistence. We prove this result in Appendix A in **Supplementary Material**.

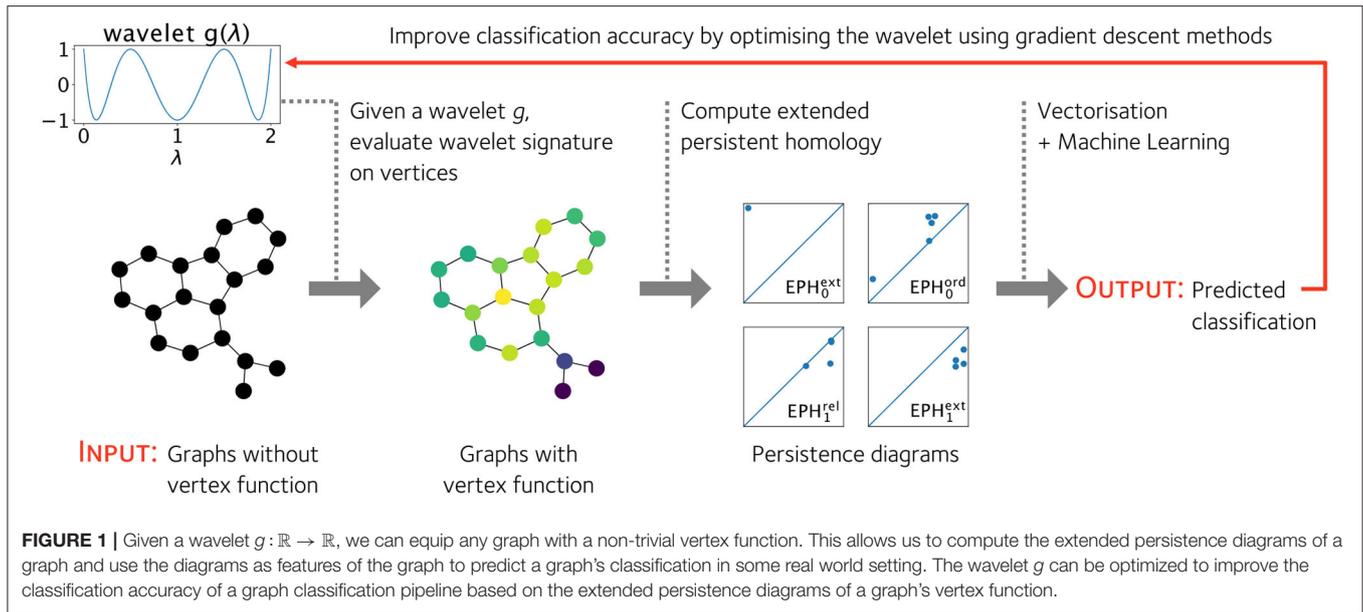
Finally, in section 4, we apply our framework to graph classification problems on several benchmark datasets. We show that our model is competitive with state-of-the-art persistence-based models. In particular, optimizing the vertex function appreciably improves the prediction accuracy on some datasets.

## 2. FILTER FUNCTION PARAMETERIZATION

We describe our recipe for assigning vertex functions to any simplicial graph  $G = (V, E)$  based on a parameterized spectral wavelet, the first part  $F$  of the loss function

$$\mathcal{L} : \mathcal{M} \xrightarrow{F} \mathbb{R}^V \xrightarrow{\text{EPH}} \mathbf{Bar}^4 \xrightarrow{\text{Out}} \mathbb{R}. \quad (\text{Equation 1 recalled})$$

Our recipe is based on a graph's wavelet signature, a vertex function derived from the graph's Laplacian. The wavelet signature also depends on a so-called 'wavelet function' in



$g: \mathbb{R} \rightarrow \mathbb{R}$ , which is independent of the graph. By modulating the wavelet, we can jointly vary the wavelet signature across many graphs. We parameterize the wavelet using a finite linear combination of basis functions, such that the wavelet signature can be manipulated in a computationally tractable way. In the following section, we define the wavelet signature and describe our linear approach to wavelet parameterization.

### 2.1. Wavelet Signatures

The wavelet signature is a vertex function initially derived from wavelet transforms of vertex functions on graphs [29], a generalization of wavelet transforms for square integrable functions on Euclidean space [30, 31] for signal analysis [32]. Wavelet signatures for graphs have been applied to encode geometric information about meshes of 3D shapes [27, 32]. Special cases of wavelets signatures, such as the heat kernel signature [33] and wave kernel signature [34], have also been applied to describe graphs and 3D shapes [35, 36].

The wavelet signature of a graph is constructed from the graph's Laplacian operator. A graph's normalized Laplacian  $L \in \mathbb{R}^{V \times V}$  is a symmetric positive semi-definite matrix, whose entries are given by

$$L_{uv} = \begin{cases} 1 & u = v \\ -\frac{1}{\sqrt{k_u k_v}} & (u, v) \in E \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where  $k_u$  is the degree of vertex  $u$ . The Laplacian's eigenvalues  $\lambda$  and eigenvectors  $\phi$  are known to encode various topological and geometric information about the graph [37, 38]; for example, the number of zero eigenvalues corresponds to the number of connected components of the graph. The spectrum of the normalized Laplacian have real eigenvalues in  $[0, 2]$  [37]. As such, any function  $g: \mathbb{R} \rightarrow \mathbb{R}$  evaluated on the eigenvalues need only

be defined on  $[0, 2]$ . Moreover, functions on a compact domain are easily parameterized using convenient bases.

**Definition 2.1.** (Wavelet Signature [27]) Let  $L \in \mathbb{R}^{V \times V}$  be the normalized Laplacian of a simplicial graph  $G = (V, E)$ . Let  $\phi_1, \dots, \phi_{|V|}$  be an orthonormal eigenbasis for  $L$  and  $\lambda_1, \dots, \lambda_{|V|}$  be their corresponding eigenvalues. The wavelet signature  $W: \mathbb{R}^{[0,2]} \rightarrow \mathbb{R}^V$  maps a function  $g: [0, 2] \rightarrow \mathbb{R}$ , which we refer to as a *wavelet*, to a vertex function  $W(g) \in \mathbb{R}^V$  linearly, where the value of  $W(g)$  on vertex  $v$  is given by

$$W(g)_v = \sum_{i=1}^{|V|} g(\lambda_i) (\phi_i)_v^2, \quad (3)$$

and  $(\phi_i)_v$  denotes the component of eigenvector  $\phi_i$  corresponding to vertex  $v$ .

If the eigenvalues of  $L$  have geometric multiplicity one (i.e., their eigenspaces are one dimensional), then the orthonormal eigenvectors are uniquely defined up to a choice of sign. It is then apparent from Equation (3) that the wavelet signature is independent of the choice of sign. However, if some eigenvalues have geometric multiplicity greater than one, then the orthonormal eigenvectors of  $L$  are uniquely defined up to orthonormal transformations in the individual eigenspaces. However, the wavelet signature is well-defined even when the multiplicities of eigenvalues are greater than one. This is the content of the next Proposition, whose proof is deferred to Appendix B in **Supplementary Material**.

**PROPOSITION 2.2.** *The wavelet signature of a graph is independent of the choice of orthonormal eigenbasis for the Laplacian.*

**Remark 2.3.** In addition to the traditional view of wavelets from a spectral signal processing perspective [29], we can also relate the

wavelet signature of a vertex  $v$  to the degrees of vertices in some neighborhood of  $v$  prescribed by  $g$ . Consider a wavelet  $g : [0, 2] \rightarrow \mathbb{R}$ . On a finite graph  $G$ , the normalized Laplacian  $L$  has at most  $|V|$  many distinct eigenvalues. As such, there exists a polynomial  $\hat{g}(x) = \sum_{n=0}^p a_n x^n$  of finite order that interpolates  $g$  at the eigenvalues  $g(\lambda_i) = \hat{g}(\lambda_i)$ . Therefore,  $W(g) = W(\hat{g})$ . Moreover, the vertex values assigned by  $W(\hat{g})$  are the diagonal entries of the matrix polynomial  $\hat{g}(L)$ :

$$\begin{aligned} \hat{g}(L)_{vv} &= \sum_{n=0}^p a_n (L^n)_{vv} = \sum_{i=1}^{|V|} \hat{g}(\lambda_i) (\phi_i)_v^2 = \sum_{i=1}^{|V|} g(\lambda_i) (\phi_i)_v^2 \\ &= W(g)_{vv}. \end{aligned} \tag{4}$$

Furthermore, we can also write the matrix polynomial  $\hat{g}(L)$  as a matrix polynomial in  $A = I - L$ , the *normalized adjacency matrix*. From the definition of  $L$ , we can compute the diagonal entry of a monomial  $A^r$  corresponding to vertex  $v$  as an inverse degree weighted count of paths<sup>1</sup>  $[v_0, v_1, \dots, v_r]$  on the graph which begin and end on vertex  $v = v_0 = v_r$  [39]:

$$(A^r)_{vv} = \frac{1}{k_v} \sum_{[v, v_1, \dots, v_{r-1}, v]} \left( \prod_{l=1}^{r-1} \frac{1}{k_{v_l}} \right). \tag{5}$$

By expressing the wavelet signature as a matrix polynomial in  $A$ , we see that  $g$  controls how information at different length scales of the graph contribute to the wavelet signature. For instance, if  $g$  were an order  $p$  polynomial, then  $W(g)_v$  only takes the degrees of vertices that are  $\lfloor p/2 \rfloor$  away from  $v$  into account. As a corollary, since  $W(g)$  can be specified by replacing  $g$  with a polynomial  $\hat{g}$  of order at most  $|V| - 1$ , the wavelet signature at a vertex is only dependent on the subgraph of  $G$  that is within  $\lfloor |V| - 1 \rfloor / 2$  steps away from  $v$ .

### 2.2. Parameterizing the Wavelet

We see from Remark 2.3 that the choice of wavelet  $g$  determines how the topology and geometry of the graph is reflected in the vertex function. Though the space of wavelets is potentially infinite dimensional, here we only consider wavelets  $g_\theta(x)$  that are parameterized by parameters  $\theta$  in a finite dimensional manifold, so that we can easily optimize them using computational methods. In particular, we focus on wavelets written as a linear combination of  $m$  *basis functions*  $h_1, \dots, h_m : [0, 2] \rightarrow \mathbb{R}$

$$g_\theta(x) := \sum_{j=1}^m \theta_j h_j(x) \tag{6}$$

This parameterization of wavelets in turn defines a parameterization of vertex functions  $F : \mathbb{R}^m \rightarrow \mathbb{R}^V$  for our optimization pipeline in Equation (1)

$$F : \theta \in \mathbb{R}^m \mapsto F(\theta) := W(g_\theta) \in \mathbb{R}^V. \tag{7}$$

<sup>1</sup>Here a path refers to a sequences of vertices that are connected to the next vertex in the sequence by an edge.

Since  $W(g)$  is a linear function of the wavelet  $g$ ,  $F$  is a linear transformation:

$$F(\theta) = W \left( \sum_{j=1}^m \theta_j h_j(x) \right) = \sum_{j=1}^m \theta_j W(h_j). \tag{8}$$

We can write  $F$  as a  $|V| \times m$  matrix acting on a vector  $[\theta_1, \dots, \theta_m]^T \in \mathbb{R}^m$ , whose columns are the vertex functions  $W(h_j)$ .

**Example 2.4** (Chebyshev Polynomials). Any Lipschitz continuous function on an interval can be well-approximated by truncating its Chebyshev series at some finite order [40]. The Chebyshev polynomials  $T_n : [-1, 1] \rightarrow \mathbb{R}$

$$T_n(x) = \cos(n \arccos(x)) \quad n \in \mathbb{N}_{\geq 0}. \tag{9}$$

form an orthonormal set of functions. We can thus consider  $h_j(\lambda) = T_j(\lambda - 1)$ ,  $j = 0, 2, \dots, m$  as a naïve basis for wavelets. We exclude  $T_1(x) = x$  in the linear combination as  $W(T_1(1 - x)) = 0$  for graphs without self loops.

**Example 2.5** (Radial Basis Functions). In the machine learning community, a *radial function* refers loosely to a continuous monotonically decreasing function  $\rho : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ . There are many possible choices for  $\rho$ , for example, the inverse multiquadric

$$\rho(r) = \left( \left( \frac{r}{\epsilon} \right)^2 + 1 \right)^{-\frac{1}{2}} \tag{10}$$

where  $\epsilon \neq 0$  is a width parameter. We can obtain a naïve wavelet basis  $h_j(x) = \rho(\|x - x_j\|)$  using copies of  $\rho$  offset by a collection of centroids  $x_j \in \mathbb{R}$  along  $\mathbb{R}$ . In general, the centroids are parameters that could be optimized, but we fix them in this study. This parameterization can be considered as a *radial basis function neural network*. RBNNs are well-studied in function approximation and subsequently machine learning; we refer readers to [41, 42] for further details.

### 2.3. The Choice of Wavelet Basis

The choice of basis functions determines the space of wavelet signatures and also the numerical stability of the basis function coefficients which serve as the wavelet signature parameters. The stability of the parameterization depends on the graphs as much as the choice of wavelet basis  $h_1, \dots, h_m$ . We can analyse the stability of a parameterization  $F$  by its the singular value decomposition

$$F = \sum_{k=1}^r \sigma_k u_k v_k^T \tag{11}$$

where  $\sigma_1, \dots, \sigma_r$  are the non-zero singular values of the matrix, and  $u_k \in \mathbb{R}^{|V|}$  and  $v_k \in \mathbb{R}^m$  are orthonormal sets of vectors, respectively. If the distribution of singular values span many orders of magnitude, we say the parameterization is *ill-conditioned*. An ill-conditioned parameterization interferes with

the convergence of gradient descent algorithms on a loss function evaluated on wavelet signatures. We discuss the relationship between the conditioning of  $F$  and the stability of gradient descent in detail in Remark 2.7.

We empirically observe that the coefficients of a naïve choice of basis functions, such as Chebyshev polynomials or radial basis functions, are numerically ill-conditioned. In **Figure A2 (Appendix in Supplementary Material.)**, we can see that the singular values of radial basis function and Chebyshev polynomial parameterizations, respectively, are distributed across a large range on the logarithmic scale for some datasets of graphs in machine learning. We address this problem by picking out a new wavelet basis

$$h'_k(x) = \frac{1}{\sigma_k} \sum_{j=1}^m (v_k)_j h_j(x), \quad k = 1, \dots, r, \quad (12)$$

where  $\sigma_k$  are the singular values of  $F$  and  $v_k$  are the associated vectors in  $\mathbb{R}^m$  from the singular value decomposition of matrix  $F$  in Equation (11). Then the parameterization  $F' : \mathbb{R}^r \rightarrow \mathbb{R}^V$

$$F'(\theta') = \sum_{k=1}^r \theta'_k W(h'_k). \quad (13)$$

have singular values equal to one, since this is a linear combination of orthonormal vectors  $u_k \in \mathbb{R}^V$ :

$$W(h'_k) = \sum_{j=1}^m \frac{1}{\sigma_k} (v_k)_j W(h_j) = \frac{1}{\sigma_k} F v_k = u_k. \quad (14)$$

As an example, we plot the new wavelet basis  $h'_k$  derived from a 12 parameter radial basis function parameterization for the MUTAG dataset in **Figure A3 in Appendix B in Supplementary Material.**

**Remark 2.6** (Learning a Wavelet Basis for Wavelet Signatures on Multiple Graphs). In the case where the wavelet coefficients parameterize the wavelet signatures over graphs  $G_1, \dots, G_N$ , we can view the maps  $F_1, \dots, F_N$  that map wavelet basis coefficients to vertex functions of graphs  $G_1, \dots, G_N$ , respectively, as a parameterization for the disjoint union  $\bigsqcup_i G_i$ :

$$f = \begin{bmatrix} f_1 \\ \vdots \\ f_N \end{bmatrix} = \begin{bmatrix} F_1 \\ \vdots \\ F_N \end{bmatrix} \theta = : F\theta. \quad (15)$$

We can then perform a singular value decomposition of the parameterization  $F$  on  $\bigsqcup_i G_i$  and derive a new, well-conditioned basis.

**Remark 2.7** (Why the Conditioning of  $F$  Matters). Let us optimize a loss function  $\mathcal{L}$  on the parameter space of wavelet coefficients  $\theta$  using a gradient descent algorithm. In a gradient descent step of step size  $s$ , the wavelet coefficients are updated to  $\theta \mapsto \theta - s \nabla_{\theta} \mathcal{L}$ . Using the singular value decomposition of  $F$  (Equation 11), we can write

$$\nabla_{\theta} \mathcal{L} = \nabla_{\theta} f^T \nabla_f \mathcal{L} = F^T \nabla_f \mathcal{L} = \sum_{k=1}^r \sigma_k \langle \nabla_f \mathcal{L}, u_k \rangle v_k. \quad (16)$$

The change in the vertex function is simply the matrix  $F$  applied to the change in wavelet parameters. Hence, the vertex function is updated to  $f \mapsto f - s F \nabla_{\theta} \mathcal{L}$ , where

$$F \nabla_{\theta} \mathcal{L} = \sum_{k=1}^r \sigma_k \langle \nabla_f \mathcal{L}, u_k \rangle F v_k = \sum_{k=1}^r \sigma_k^2 \langle \nabla_f \mathcal{L}, u_k \rangle u_k. \quad (17)$$

If the loss function  $\mathcal{L}$  has large second derivatives— for example, due to non-linearities in the function on persistence diagrams **Out : Bar<sup>4</sup>  $\rightarrow$   $\mathbb{R}$** —the projections  $\langle \nabla_f \mathcal{L}, u_k \rangle$  in Equations (16) and (17) may change dramatically from one gradient descent update to another. If the smallest singular value is much smaller than the largest, then updates to the wavelet signature can be especially unstable throughout the optimization process. This source of instability can be removed if we choose a parameterization with uniform singular values  $\sigma_k = 1$ . In this case, the update to  $f$  is simply the projection of  $\nabla_f \mathcal{L}$  onto the space of wavelet signatures spanned by  $u_1, \dots, u_r$ , without any distortion introduced by non-uniform singular values:

$$f \mapsto f - s \sum_{k=1}^r \langle u_k, \nabla_f \mathcal{L} \rangle u_k. \quad (18)$$

### 3. EXTENDED PERSISTENT HOMOLOGY

The homology of a given graph is a computable vector space whose dimension counts the number of connected components or cycles in the graph. Finer information can be retained by filtering the graph and analyzing the evolution of the homology throughout the filtration. This evolution is described by a set of *extended persistence diagrams* (a.k.a. *extended barcodes*), a multiset of points  $\langle b, d \rangle$  that record the birth  $b$  and death of homological features in the filtration. In this section, we begin by summarizing these constructions. We refer the reader to Zomorodian and Carlsson [4], Edelsbrunner and Harer [5], and Cohen-Steiner et al. [43] for full treatments of the theory of Persistence.

Compared to *ordinary persistence*, extended persistence is a more informative and convenient feature map for graphs. Extended persistence encodes strictly more information than ordinary persistence. For instance, the cycles of a graph are represented as points with  $d = \infty$  in ordinary persistence. Thus, only the birth coordinate  $b$  of such points contain useful information about the cycles. In contrast, the corresponding points in extended persistence are each endowed with a finite death time  $d$ , thus associating extra information to the cycles. The points at infinity in ordinary persistence also introduce obstacles to vectorization procedures, as often arbitrary finite cutoffs are needed to ‘tame’ the persistence diagrams before vectorization.

#### 3.1. Extended Persistent Homology

Let  $G = (V, E)$  be a finite graph without double edges and self-loops. For the purposes of this paper, the associated *extended persistent homology* is a map

$$\text{EPH} : \mathbb{R}^V \rightarrow \text{Bar}^4$$

from functions  $f \in \mathbb{R}^V$  on its vertices to the space of four persistence diagrams or barcodes, which we define below. The map arises from a filtration of the graph, a sequential attachment of vertices and edges in ascending or descending order of  $f$ . We extend  $f$  on each edge  $e = (v, v')$  by the maximal value of  $f$  over the vertices  $v$  and  $v'$ , and we then let  $G_t \subset G$  be the sub graph induced by vertices taking value less than  $t$ . Then we have the following sequence of inclusions:

$$\emptyset \longrightarrow \dots \longrightarrow G_s \xrightarrow{s \leq t} G_t \longrightarrow \dots \longrightarrow G. \quad (19)$$

Similarly, the sub graphs  $G^t \subset G$  induced by vertices taking value greater than  $t$  assemble into a sequence of inclusions:

$$G \longleftarrow \dots \longleftarrow G^s \xleftarrow{s \leq t} G^t \longleftarrow \dots \longleftarrow \emptyset. \quad (20)$$

The changes in the topology of the graph along the filtration in ascending and descending order of  $f$  can be detected by its extended persistence module, indexed over the poset  $\mathbb{R} \cup \{\infty\} \cup \mathbb{R}^{\text{op}}$ :

$$\begin{array}{ccccccc} 0 = H_p(\emptyset) & \longrightarrow & \dots & \longrightarrow & H_p(G_s) & \xrightarrow{s \leq t} & H_p(G_t) & \longrightarrow & \dots & \longrightarrow & H_p(G) \\ V_p(f) : & & & & & & & & & & \downarrow \cong \\ 0 = H_p(G, G) & \longleftarrow & \dots & \longleftarrow & H_p(G, G^s) & \xleftarrow{s \leq t} & H_p(G, G^t) & \longleftarrow & \dots & \longleftarrow & H_p(G, \emptyset) \end{array} \quad (21)$$

where  $H_p$  is the singular (relative) homology functor in degree  $p \in 0, 1$  with coefficients in a fixed field, chosen to be  $\mathbb{Z}/2\mathbb{Z}$  in practice. In general terms, the modules  $V_0(f)$  and  $V_1(f)$  together capture the evolution of the connected components and loops in the sub graphs of  $G$  induced by the function  $f$ .

Each module  $V_p(f)$  is completely characterized by a finite multi-set  $\text{EPH}_p(f)$  of pairs of real numbers  $\langle b, d \rangle$  called intervals representing the birth and death of homological features. Following Cohen-Steiner et al. [44], the intervals in  $\text{EPH}_p(f)$  are further partitioned according to the type of homological feature they represent:

$$\begin{aligned} \text{EPH}_p(f) = & \underbrace{\{\langle b, d \rangle \mid b < d < \infty\}}_{=\text{EPH}_p^{\text{ord}}(f)} \sqcup \underbrace{\{\langle b, d \rangle \mid b < \infty < d\}}_{=\text{EPH}_p^{\text{ext}}(f)} \\ & \sqcup \underbrace{\{\langle b, d \rangle \mid \infty < b < d\}}_{=\text{EPH}_p^{\text{rel}}(f)}. \end{aligned} \quad (22)$$

Each of the three finite multiset  $\text{EPH}_p^k(f)$ , for  $k \in \{\text{ord}, \text{ext}, \text{rel}\}$ , is an element in the space  $\mathbf{Bar}$  of so-called barcodes or persistence diagrams. However,  $\text{EPH}_0^{\text{rel}}(f)$  and  $\text{EPH}_1^{\text{ord}}(f)$  being trivial for graphs, we refer to the collection of four remaining persistence diagrams

$$\text{EPH}(f) = \left[ \text{EPH}_0^{\text{ord}}(f), \text{EPH}_0^{\text{ext}}(f), \text{EPH}_1^{\text{ext}}(f), \text{EPH}_1^{\text{rel}}(f) \right] \in \mathbf{Bar}^4 \quad (23)$$

as the extended barcode or extended persistence diagram of  $f$ . We have thus defined the extended persistence map

$$\text{EPH} : \mathbb{R}^V \rightarrow \mathbf{Bar}^4.$$

**Remark 3.1.** If we only apply homology to the filtration of Equation (19), we get an ordinary persistence module indexed over the real line, which is essentially the first row in Equation (21). This module is characterized by a unique barcode  $\text{PH}_p(f) \in \mathbf{Bar}$ . We refer to the map

$$\text{PH} : f \in \mathbb{R}^V \mapsto [\text{PH}_0(f), \text{PH}_1(f)] \in \mathbf{Bar}^2 \quad (24)$$

as the ordinary persistence map.

### 3.2. Differentiability of Extended Persistence

The extended persistence map can be shown to be locally Lipschitz by the Stability theorem [44]. The Rademacher theorem states that any real-valued function that is locally Lipschitz is differentiable on a full measure set. Thus, so is our loss function

$$\mathcal{L} : \mathcal{M} \xrightarrow{F} \mathbb{R}^V \xrightarrow{\text{EPH}} \mathbf{Bar}^4 \xrightarrow{\text{Out}} \mathbb{R}. \quad (\text{Equation 1 recalled})$$

as long as  $\text{Out}$  and  $F$  are smooth or locally Lipschitz<sup>2</sup>. If a loss function  $\mathcal{L}$  is locally Lipschitz, we can use stochastic gradient descent as a paradigm for optimization. Nonetheless, the theorem above does not rule out dense sets of non differentiability in general.

In this section, we show that the set where  $\text{EPH}$  is not differentiable is not pathological. Namely, we show that  $\text{EPH}$  is generically differentiable, i.e., differentiable on an open dense subset. This property guarantees that local gradients yield reliable descent directions in a neighborhood of the current iterate. We recall from Leygonie et al. [21] the definition of differentiability for maps to barcodes.

We call a map  $F : \mathcal{M} \rightarrow \mathbb{R}^V$  a parameterization, as it corresponds to a selection of filter functions over  $G$  parameterized by the manifold  $\mathcal{M}$ . Then  $B := \text{EPH} \circ F$  is the barcode valued map whose differentiability properties are of interest in applications.

**Definition 3.2.** A map  $B : \mathcal{M} \rightarrow \mathbf{Bar}$  on a smooth manifold  $\mathcal{M}$  is said to be differentiable at  $\theta \in \mathcal{M}$  if for some neighborhood  $U$  of  $\theta$ , there exists a finite collection of differentiable maps<sup>3</sup>  $b_i, d_i : U \rightarrow \mathbb{R} \cup \{\infty\}$ , called a local coordinate system for  $B$  at  $\theta$ , such that

$$\forall \theta' \in U, B(\theta') = \{\langle b_i(\theta'), d_i(\theta') \rangle \mid b_i(\theta') \neq d_i(\theta')\}.$$

For  $N \in \mathbb{N}$ , we say that a map  $B : \mathcal{M} \rightarrow \mathbf{Bar}^N$  is differentiable at  $\theta$  if all its components are so.

In Leygonie et al. [21], it is proven that the composition  $\text{PH} \circ F$  is generically differentiable as long as  $F$  is so. It is possible to show that  $\text{EPH} \circ F$  is generically differentiable along the

<sup>2</sup>In practice, a locally Lipschitz  $\text{Out}$  can be constructed out of Lipschitz stable vectorization methods, such as Persistence Landscapes [45] and Persistence Images [46].

<sup>3</sup>By convention, a differentiable map that takes the value  $\infty$  is constant.

same lines, but we rather provide an alternative argument in the Appendix. Namely, we rely on the fact that the extended persistence of  $G$  can be decoded from the ordinary persistence of the cone complex  $C(G)$ , a connection first noted in Cohen-Steiner et al. [44] for computational purposes.

**PROPOSITION 3.3.** *Let  $F: \mathcal{M} \rightarrow \mathbb{R}^V$  be a generically differentiable parameterization. Then the composition  $\text{EPH} \circ F$  is generically differentiable.*

For completeness, the proof provided in the Appendix treats the general case of a finite simplicial complex  $K$  of arbitrary dimension.

## 4. BINARY GRAPH CLASSIFICATION

We investigate whether optimizing the extended persistence of wavelet signatures can be usefully applied to graph classification problems, where persistence diagrams are used as features to predict discrete, real life attributes of networks. In this setting, we aim to learn  $\theta \in \mathcal{M}$  that minimize the classification error of graphs over a training dataset.

We apply our wavelet optimization framework to classification problems on the graph datasets MUTAG [47, 48], COX2 [49], DHFR [49], NCI1 [50, 51], PROTEINS [52, 53], and IMDB-B [54]. The former five datasets are biochemical molecules and IMDB-B is a collection of social ego networks. In our models, we use persistence images [46] as a fixed vectorization method and use a feed forward neural network to map the persistence images to classification labels. We also include the eigenvalues of the graph Laplacian as additional features; model particulars are described in the sections below.

To illustrate the effect of wavelet optimization on different classification problems, we also perform a set of *control experiments* where for the same model architecture, we fix the wavelet and only optimize the parameters of the neural network. The control experiment functions as a baseline against which we assess the efficacy of wavelet optimization.

We benchmark our results with two existing persistence based architectures, PersLay [7] and GFL [26]. PersLay optimizes the vectorization parameters and use two heat kernel signatures as fixed rather than optimizable vertex functions for computing extended persistence. GFL optimizes and parameterizes vertex functions using a graph isomorphism network [1], and computes *ordinary* sublevel and superlevel set persistence instead of extended persistence.

### 4.1. Model Architecture

We give a high level description of our model and relegate details and hyperparameter choices of the vectorization method and neural network architecture to Appendix C in **Supplementary Material**. In our setting, the extended persistence diagrams of the optimizable wavelet signatures for each graph are vectorized as persistence images. We also include the static persistence images of a *fixed* heat kernel signature,  $W(e^{-0.1x})$ , as an additional set of features, alongside some non-persistence features. Both the optimized and static persistence diagrams are transformed into the persistence images using identical hyperparameters. We feed the optimizable and

**TABLE 1** | Binary classification accuracy of our model where we vary whether non-Persistence features are included and whether the wavelet is optimized.

	Persistence only		Non-persistence features incl.	
	Control	Wavelet Opt.	Control	Wavelet Opt.
MUTAG	89.2 ± 0.6	89.8 ± 0.8	89.0 ± 0.6	90.4 ± 0.4
COX2	79.6 ± 1.0	79.4 ± 0.7	80.8 ± 1.0	80.8 ± 1.0
DHFR	79.9 ± 0.4	80.4 ± 0.4	80.3 ± 0.9	81.0 ± 0.9
NCI1	73.7 ± 0.2	74.3 ± 0.5	74.3 ± 0.3	74.4 ± 0.3
PROTEINS	72.9 ± 0.3	73.0 ± 0.4	74.5 ± 0.4	74.6 ± 0.6
IMDB-B	68.3 ± 0.5	68.6 ± 0.7	71.6 ± 0.9	72.0 ± 0.7

*The reported accuracies are the mean over 10 ten-folds, recorded at epochs reported in Table C1. We also provide standard deviations of the 10 mean accuracies of each ten-fold. See section 4.1.2 for the particulars about the non-persistence features.*

static persistence images into two separate convolutional neural networks (CNNs) with the same architecture. Similarly, we feed the non-persistence features as a vector into a separate multilayer perceptron. The outputs of the CNNs are concatenated with the outputs of the multi-layer perceptron. Finally, an affine transformation sends the concatenated vector to a real number whose sign determines the binary classification.

#### 4.1.1. Wavelet Parameterization

We choose a space of wavelets spanned by 12 inverse multiquadric radial basis functions

$$h_j(x) = \left( \left( \frac{x - x_j}{\epsilon} \right)^2 + 1 \right)^{-\frac{1}{2}} \quad (25)$$

whose centroids  $x_j$  are located at  $x_j = 2(j - 1)/9$ ,  $j = 0, \dots, 11$ . The width parameter is chosen to be the distance between the centroids,  $\epsilon = 2/9$ . On each dataset, we derive a numerically stable parameterization using the procedure described in section 2.2; the parameters we optimize are the coefficients of the new basis given by Equation (12). We initialize the parameters by fitting them via least squares to the heat kernel signature  $W(e^{-10x})$  on the whole dataset of graphs.

#### 4.1.2. Non-Persistence Features

We also incorporate the eigenvalues of the normalized Laplacian as additional, fixed features of the graph. Since the number of eigenvalues for a given graph is equal to the number of vertices, it differs between graphs in the same dataset. To encode the information represented in the eigenvalues as a fixed length vector, we first sort the eigenvalues into a time-series; we then compute the log path signature of the time series up to level four, which is a fixed length vector in  $\mathbb{R}^8$ . The log-signature captures the geometric features of the path; we refer readers to Chevyrev and Kormilitzin [55] for details about path signatures. For IMDB-B in particular, we also include the maxima and minima of the heat kernel signatures  $W(e^{-10x})$  and  $W(e^{-0.1x})$ , respectively, of each graph.

## 4.2. Experimental Set Up

We employ a *10 ten-fold* test-train split scheme on each dataset to measure the accuracy of our model. Each ten-fold is a set of ten

**TABLE 2** | Binary classification accuracy on datasets of graphs.

Node attr.	Non-persistence state-of-the-art				Persistence based					
	P-SAN	RetGK	GIN	FGSD	PWL	GFL	Perslay	Control	Wavelet Opt.	
	[57]	[58]	[1]	[59]	[8]	[26]	[7]	This paper		
	Yes	No		Yes	No					
MUTAG	92.6	90.3 ± 1.1	89.4	92.1	90.5 ± 1.3	–	–	89.8 ± 0.9	89.0±0.6	<b>90.4±1.3</b>
COX2	–	81.4 ± 0.6	–	–	–	–	–	<b>80.9 ± 1.0</b>	80.8±0.4	80.8±1.0
DHFR	–	82.5 ± 0.8	–	–	–	–	–	80.3 ± 0.8	80.0±0.4	<b>81.0±0.9</b>
NCI1	78.6	84.5 ± 0.2	82.7	79.8	85.6 ± 0.3	77.2	71.2	73.5 ± 0.3	74.3±0.3	<b>74.4±0.3</b>
PROTEINS	75.9	78.0 ± 0.3	76.2	73.4	75.9 ± 0.8	73.4	74.1	<b>74.8 ± 0.3</b>	74.5±0.4	74.6±0.6
IMDB-B	71.0	72.3 ± 0.6	75.1	73.6	73.0 ± 1.0	–	<b>74.5</b>	71.2 ± 0.7	71.6±0.9	72.0±0.7
# Ten-folds	10	10	1	1	10	1	1	10	10	10

The best accuracy of persistence-based models without using node attributes is made bold for each dataset. The performance of our model is reported in the column Wavelet Opt. on the right hand side. The accuracies of the control model, where the wavelet parameters are fixed to the initial values, are shown in the column Control. Both these models use additional features (see section 4.1.2). The accuracies of our model are the means over 10 ten-folds, recorded at epochs reported in **Table C1**. We also provide the standard deviations of the 10 mean accuracies of each ten-fold. For other architectures, we indicate whether their accuracies were reported as averages over 1 ten-fold or 10 ten-fold in the bottom row of the table. To avoid confusion, we leave out the errors reported for P-SAN, GIN and GFL and refer the reader to the original sources, as they were calculated using a different formula. Errors were not reported in [59] for FGSD.

experiments, corresponding to a random partition of the dataset into ten portions. In each experiment, a different portion is selected as the test set while the model is trained on the remaining nine portions. We perform 10 ten-folds to obtain a total of  $10 \times 10$  experiments, and report the accuracy of the classifier as the average accuracy over 100 such experiments. The epochs at which the accuracies were measured are specified in **Table C1**.

Across all experiments, we use binary cross entropy as the loss function. We use the Adam optimizer [56] with learning rate  $lr = 1e-3$  to optimize the parameters of the neural network. The wavelet parameters are updated using stochastic gradient descent with learning rate  $lr = 1e-2$ , for all datasets except for IMDB-B, where the learning rate is set to  $lr = 1e-1$ . The batch sizes for each experiment are shown in **Table C2**. In all experiments, we stop the optimization of wavelet parameters at epoch 50 while the neural network parameters continue to be optimized.

We use the GUDHI library to compute persistence, and make use of the optimization and machine learning library PyTorch for the construction of the graph classifications models.

### 4.3. Results and Discussion

In **Table 1**, we present the classification accuracies of our model. For each dataset, we perform four experiments using our model, varying whether the wavelet parameter is optimized and whether additional features are included. In **Table 2**, we show the test accuracy of our model alongside two persistence-based graph classification architectures, Perslay and GFL, as well as other state-of-the-art graph classification architectures.

We first compare the performances of our model between cases where we optimize and fix the wavelets. In **Table 1**, we see that on MUTAG and DHFR, optimizing the wavelet improves the classification accuracy regardless of whether extra features are included. On NCI1, wavelet optimization

improves the classification accuracy only persistence features are included. When we include non-persistence features in the model, the performances of the optimized and control models are statistically indistinguishable for NCI1, suggesting that the non-persistence features play a more significant role in the classification. As for COX2, PROTEINS, and IMDB-B, optimizing the wavelet coefficients do not bring about statistically significant improvements. This indicates that the initial wavelet signature—the heat kernel signature  $W(e^{-10x})$ —is a locally optimal choice of wavelet for our neural network classifier.

We now compare our architecture to other persistence based architectures, Perslay and GFL, where node attributes are excluded from their vertex function models. Except on PROTEINS, our wavelet optimized model matches or exceeds Perslay. While our model architecture and choice of wavelet initialization is similar to that of Perslay, we differ in two important respects. Perslay fixes the vertex functions but optimizes the weights assigned to points on the persistence diagrams, as well as the parameters of the persistence images. Our improvements on Perslay for MUTAG, DHFR, and IMDB-B indicate that vertex function optimization yields improvements that cannot be obtained through vectorization optimization alone on some datasets of graphs.

Compared to GFL (without node attributes), both Perslay and our architecture achieves similar or higher classification accuracies on PROTEINS and NCI1. This supports wavelet signatures being viable models for vertex functions on those datasets. On the other hand, both Perslay and our model lag behind GFL on IMDB-B. We attribute this to the fact that IMDB-B, unlike the other bioinformatics datasets, consists of densely connected graphs. The graphs in IMDB-B have diameter at most two and 14% of the graphs are cliques. This fact has two consequences. First, we expect the one-layer GIN used in GFL—a

local topology summary—to be highly effective in optimizing for the salient features of a graph with small diameter. Second, the extended persistence modules for cliques have zero persistence, since all vertices are assigned the same function value due to symmetry. In contrast, ordinary persistence used in GFL is able to capture the cycles in a complete graph as points with infinite persistence.

Compared to non-persistence state-of-the-art architectures in **Table 1**, our model achieves competitive accuracies on MUTAG, COX2, and DHFR. For NC11 and PROTEINS, all persistence architectures listed that exclude additional node attributes perform poorly in comparison, though PWL was able to achieve leading results with node attributes.

All in all, we observe that wavelet signatures can be an effective parameterization of vertex functions when we use extended persistence as features for graph classification. In particular, on some bioinformatics datasets, we show that optimizing the wavelet signature can lead to improvements in classification accuracy. The wavelet signature approach is complementary to the GFL approach to vertex function parameterization as they show strengths on different datasets.

## 5. CONCLUSION

We have presented a framework for equipping any graph  $G$  with a set of extended persistence diagrams  $\text{EPH} \circ F: \mathcal{M} \rightarrow \text{Bar}^4$  parameterized over a manifold  $\mathcal{M}$ , a parameter space for the graph's wavelet signature. We described how wavelet signatures can be parameterized and interpreted. Given a function on extended persistence diagrams  $\text{Out}: \text{Bar}^4 \rightarrow \mathbb{R}$  that is differentiable, we have shown how a loss function  $\mathcal{L} = \text{Out} \circ \text{EPH} \circ F$  can be generically differentiable with respect to  $\theta \in \mathcal{M}$  as  $\mathcal{L}$ . Thus, we can apply gradient descent methods to optimize the extended persistence diagrams of a graph to minimize  $\mathcal{L}$ .

We applied this framework to a graph classification architecture where the wavelet signature is optimized for classification accuracy. We are able to demonstrate an increase in accuracy on several benchmark datasets where the wavelet is optimized, and perform competitively with state-of-the-art persistence based graph classification architectures.

## REFERENCES

- Xu B, Shen H, Cao Q, Qiu Y, Cheng X. Graph wavelet neural network. *arXiv [Preprint]*. (2019) arXiv:1904.07785.
- Vishwanathan SVN, Schraudolph NN, Kondor R, Borgwardt KM. Graph kernels. *J Mach Learn. Res.* (2010) 11:1201–42.
- Shervashidze N, Vishwanathan SVN, Petri T, Mehlhorn K, Borgwardt K. Efficient graphlet kernels for large graph comparison. In: van Dyk D, Welling M, editors. *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics, Vol. 5*. Clearwater Beach, FL: PMLR (2009). p. 488–95. Available online at: <http://proceedings.mlr.press/v5/shervashidze09a.html>

## DATA AVAILABILITY STATEMENT

The code for the computational experiments in section 4 can be found in the GitHub repository [https://github.com/kmyim/Persistence\\_Opt\\_Spectral\\_Wavelets](https://github.com/kmyim/Persistence_Opt_Spectral_Wavelets). The datasets we use are publicly available at the repository TUDatasets <https://chrsmrrs.github.io/datasets/> [60].

## AUTHOR CONTRIBUTIONS

The overall framework was jointly conceived by both authors. KY was responsible for developing wavelet signatures as a vertex function parameterization framework, along with the experimental design and analysis. The proof of the differentiability of extended persistence is due to JL. Both authors participated in the writing of the article.

## FUNDING

KY was funded by the EPSRC Centre For Doctoral Training in Industrially Focused Mathematical Modelling (EP/L015803/1) with industrial sponsorship from Elsevier. JL was funded by the EPSRC grant EP/R513295/1. Both authors are members of the Centre for Topological Data Analysis, which is supported by the EPSRC grant New Approaches to Data Science: Application Driven Topological Data Analysis EP/R018472/1.

## ACKNOWLEDGMENTS

The authors would like to thank Ulrike Tillmann and Heather Harrington for their close guidance and thoughtful advice on this project. In addition, the authors would like to thank Vedit Nanda, Peter Grindrod CBE, Andrew Mellor, Steve Oudot, Mathieu Carrière, and Theo Lacombe for fruitful discussions on this subject. Finally, we are indebted to the reviewers for their thoughtful and constructive comments, which led to many improvements of the paper.

## SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/fams.2021.651467/full#supplementary-material>

- Zomorodian A, Carlsson G. Computing persistent homology. *Discr Comput Geom.* (2005) 33:249–74. doi: 10.1007/s00454-004-1146-y
- Edelsbrunner H, Harer J. Persistent homology—a survey. *Contemp Mathe.* (2008) 453:257–82. doi: 10.1090/conm/453/08802
- Zhao Q, Wang Y. Learning metrics for persistence-based summaries and applications for graph classification. In: Wallach H, Larochelle H, Beygelzimer A, dAlché-Buc F, Fox E, Garnett R, editors. *Advances in Neural Information Processing Systems, Vol. 32*. Red Hook, NY: Curran Associates, Inc. (2019). p. 9859–70. Available online at: <https://proceedings.neurips.cc/paper/2019/file/12780ea688a71dabc284b064add459a4-Paper.pdf>
- Carrière M, Chazal F, Ike Y, Lacombe T, Royer M, and Umeda Y. Perslay: a neural network layer for persistence diagrams and new graph topological

- signatures. In: Chiappa S, Calandra R, editors. *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, Vol. 108. PMLR (2020). p. 2786–96. Available online at: <http://proceedings.mlr.press/v108/carriere20a.html>.
8. Rieck B, Bock C, Borgwardt K. A persistent weisfeiler-lehman procedure for graph classification. In: *International Conference on Machine Learning*. PMLR (2019). p. 5448–58.
  9. Turner K, Mileyko Y, Mukherjee S, Harer J. Fréchet means for distributions of persistence diagrams. *Discr Comput Geom.* (2014) 52:44–70. doi: 10.1007/s00454-014-9604-7
  10. Gameiro M, Hiraoka Y, Obayashi I. Continuation of point clouds via persistence diagrams. *Phys D.* (2016) 334:118–32. doi: 10.1016/j.physd.2015.11.011
  11. Oudot S, Solomon E. Inverse problems in topological persistence. In: Baas NA, Carlsson GE, Quick G, Szymik M, Thaulé M, editors. *Topological Data Analysis*. Cham: Springer (2020). p. 405–33. doi: 10.1007/978-3-030-43408-3\_16
  12. Brül-Gabrielsson R, Ganapathi-Subramanian V, Skraba P, Guibas LJ. Topology-aware surface reconstruction for point clouds. *Comp Graph Forum.* (2020) 39:197–207. doi: 10.1111/cgf.14079
  13. Poulénard A, Skraba P, Ovsjanikov M. Topological function optimization for continuous shape matching. *Comp Graph Forum.* (2018) 37:13–25. doi: 10.1111/cgf.13487
  14. Moor M, Horn M, Rieck B, Borgwardt K. Topological autoencoders. In: *International Conference on Machine Learning*. PMLR (2020). p. 7045–54.
  15. Hofer C, Kwitt R, Niethammer M, Dixit M. Connectivity-optimized representation learning via persistent homology. In: Chaudhuri K, Salakhutdinov R, editors. *Proceedings of the 36th International Conference on Machine Learning*. Long Beach, CA: PMLR (2019). p. 2751–60. Available online at: <http://proceedings.mlr.press/v97/hofer19a.html>
  16. Gabrielsson RB, Nelson BJ, Dwarknath A, Skraba P. A topology layer for machine learning. In: *International Conference on Artificial Intelligence and Statistics*. PMLR (2020). p. 1553–63.
  17. Hu X, Li F, Samaras D, Chen C. Topology-preserving deep image segmentation. In: *Advances in Neural Information Processing Systems*. Red Hook, NY: Curran Associates, Inc. (2019). p. 5658–69. Available online at: <https://proceedings.neurips.cc/paper/2019/file/12780ea688a71dabc284b064add459a4-Paper.pdf>
  18. Clough JR, Oksuz I, Byrnes N, Schnabel JA, King AP. Explicit topological priors for deep-learning based image segmentation using persistent homology. In: *International Conference on Information Processing in Medical Imaging*. Hong Kong: Springer (2019). p. 16–28. doi: 10.1007/978-3-030-20351-1\_2
  19. Chen C, Ni X, Bai Q, Wang Y. A topological regularizer for classifiers via persistent homology. In: *The 22nd International Conference on Artificial Intelligence and Statistics*. Naha (2019). p. 2573–82.
  20. Kachan O. Persistent homology-based projection pursuit. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. (2020). p. 856–7. doi: 10.1109/CVPRW50498.2020.00436
  21. Leygonie J, Oudot S, Tillmann U. A framework for differential calculus on persistence barcodes. *arXiv [Preprint]*. (2019) arXiv:1910.00960.
  22. Carriere M, Chazal F, Glisse M, Ike Y, Kannan H. A note on stochastic subgradient descent for persistence-based functionals: convergence and practical aspects. *arXiv preprint arXiv:201008356*. (2020).
  23. Davis D, Drusvyatskiy D, Kakade S, Lee JD. Stochastic subgradient method converges on tame functions. *Found Comput Math.* (2020) 20:119–54. doi: 10.1007/s10208-018-09409-5
  24. Solomon E, Wagner A, Bendich P. A fast and robust method for global topological functional optimization. *arXiv [Preprint]*. (2020) arXiv:2009.08496.
  25. Corcoran P, Deng B. Regularization of persistent homology gradient computation. *arXiv [Preprint]*. (2020) arXiv:2011.05804.
  26. Hofer C, Graf F, Rieck B, Niethammer M, Kwitt R. Graph filtration learning. In: Daumé III H, and Singh A, editors. *Proceedings of the 37th International Conference on Machine Learning*. PMLR (2020). p. 4314–23. Available online at: <http://proceedings.mlr.press/v119/hofer20b.html>
  27. Li C, Hamza AB. A multiresolution descriptor for deformable 3D shape retrieval. *Visu Comp.* (2013) 29:513–524. doi: 10.1007/s00371-013-0815-3
  28. Rustamov RM, Guibas LJ. Wavelets on graphs via deep learning. In: Stanković L, Sejdic E, editors. *Vertex-Frequency Analysis of Graph Signals*. Cham: Springer (2019). p. 207–22. doi: 10.1007/978-3-030-03574-7
  29. Hammond DK, Vandergheynst P, Gribonval R. Wavelets on graphs via spectral graph theory. *Appl Comput Harm Anal.* (2011) 30:129–50. doi: 10.1016/j.acha.2010.04.005
  30. Graps A. An introduction to wavelets. *IEEE Comput Sci Eng.* (1995) 2:50–61. doi: 10.1109/99.388960
  31. Chui CK, Chan AK, Liu SJ. *An Introduction to Wavelets*. San Diego, CA: Academic Press (1992).
  32. Akansu AN, Haddad RA. *Multiresolution Signal Decomposition: Transforms, Subbands, and Wavelets, 2nd Edn.* San Diego, CA: Academic Press (2001). doi: 10.1016/B978-012047141-6/50002-1
  33. Sun J, Ovsjanikov M, Guibas L. A concise and provably informative multi-scale signature based on heat diffusion. *Comp Graph Forum.* (2009) 28:1383–92. doi: 10.1111/j.1467-8659.2009.01515.x
  34. Aubry M, Schlickewei U, Cremers D. The wave kernel signature: a quantum mechanical approach to shape analysis. In: *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*. IEEE (2011). p. 1626–33. doi: 10.1109/ICCVW.2011.6130444
  35. Bronstein MM, Kokkinos I. Scale-invariant heat kernel signatures for non-rigid shape recognition. In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. San Francisco, CA: IEEE (2010). p. 1704–11. doi: 10.1109/CVPR.2010.5539838
  36. Hu N, Rustamov RM, Guibas L. Stable and informative spectral signatures for graph matching. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Columbus, OH: IEEE (2014). p. 2305–12. doi: 10.1109/CVPR.2014.296
  37. Chung FRK, Graham FC. *Spectral Graph Theory*. Providence, RI: American Mathematical Society (1997).
  38. Biyikoglu, Leydold T. J., Stadler PF. *Laplacian Eigenvectors of Graphs: Perron-Frobenius and Faber-Krahn Type Theorems*. Berlin: Springer (2007). doi: 10.1007/978-3-540-73510-6
  39. Newman M. *Networks*. Oxford University Press (2018). doi: 10.1093/oso/9780198805090.001.0001
  40. Trefethen LN, Bau D III. *Numerical Linear Algebra*. Vol. 50. SIAM (1997). doi: 10.1137/1.9780898719574
  41. Chen S, Cowan CFN, Grant PM. Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Trans Neural Netw.* (1991) 2:302–9. doi: 10.1109/72.80341
  42. Park J, Sandberg TW. Universal approximation using radial-basis-function networks. *Neural Comput.* (1991) 3:246–57. doi: 10.1162/neco.1991.3.2.246
  43. Cohen-Steiner D, Edelsbrunner H, Harer J. Stability of persistence diagrams. *Discr Comput Geom.* (2007) 37:103–20. doi: 10.1007/s00454-006-1276-5
  44. Cohen-Steiner D, Edelsbrunner H, Harer J. Extending persistence using Poincaré and Lefschetz duality. *Found Comput Math.* (2009) 9:79–103. doi: 10.1007/s10208-008-9027-z
  45. Bubenik P. Statistical topological data analysis using persistence landscapes. *J Mach Learn Res.* (2015) 16:77–102.
  46. Adams H, Emerson T, Kirby M, Neville R, Peterson C, Shipman P, et al. Persistence images: a stable vector representation of persistent homology. *J Mach Learn Res.* (2017) 18:218–52.
  47. Debnath AK, Lopez de Compadre RL, Debnath G, Shusterman AJ, Hansch C. Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. Correlation with molecular orbital energies and hydrophobicity. *J Med Chem.* (1991) 34:786–97. doi: 10.1021/jm00106a046
  48. Kriege N, Mutzel P. Subgraph matching kernels for attributed graphs. In: *Proceedings of the 29th International Conference on International Conference on Machine Learning*. Madison, WI: Omnipress (2012). p. 291–8.
  49. Sutherland JJ, O'Brien LA, Weaver DF. Spline-fitting with a genetic algorithm: A method for developing classification structure- activity relationships. *J Chem Inf Comp Sci.* (2003) 43:1906–15. doi: 10.1021/ci034143r
  50. Wale N, Watson IA, Karypis G. Comparison of descriptor spaces for chemical compound retrieval and classification. *Knowl Inf Syst.* (2008) 14:347–75. doi: 10.1007/s10115-007-0103-5
  51. Shervashidze N, Schweitzer P, Van Leeuwen EJ, Mehlhorn K, Borgwardt KM. Weisfeiler-lehman graph kernels. *J Mach Learn Res.* (2011) 12:2539–61.

52. Borgwardt KM, Ong CS, Schönauer S, Vishwanathan SVN, Smola AJ, Kriegel HP. Protein function prediction via graph kernels. *Bioinformatics*. (2005) 21:i47–i56. doi: 10.1093/bioinformatics/bti1007
53. Dobson PD, Doig AJ. Distinguishing enzyme structures from non-enzymes without alignments. *J Mol Biol*. (2003) 330:771–83. doi: 10.1016/S0022-2836(03)00628-4
54. Yanardag P, Vishwanathan SVN. Deep graph kernels. in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Sydney, NSW (2015). p. 1365–74. doi: 10.1145/2783258.2783417
55. Chevyrev I, Kormilitzin A. A primer on the signature method in machine learning. *arXiv [Preprint]*. (2016) arXiv:1603.03788.
56. Kingma DP, Ba J. Adam: a method for stochastic optimization. *arXiv [Preprint]*. (2014) arXiv:1412.6980.
57. Niepert M, Ahmed M, Kutzkov K. Learning convolutional neural networks for graphs. In Balcan MF, Weinberger QK, editors. *Proceedings of The 33rd International Conference on Machine Learning*. Vol.48. New York, NY: PMLR (2016). p. 2014–23. Available online at: <http://proceedings.mlr.press/v48/niepert16.html>
58. Zhang Z, Wang M, Xiang Y, Huang Y, Nehorai A. RetGK: graph kernels based on return probabilities of random walks. In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. Red Hook, NY: Curran Associates Inc. (2018). p. 3968–78.
59. Verma S, Zhang ZL. Hunt for the unique, stable, sparse and fast feature learning on graphs. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Red Hook, NY: Curran Associates Inc. (2017). p. 87–97.
60. Morris C, Kriege NM, Bause F, Kersting K, Mutzel P, Neumann M. TUDataset: a collection of benchmark datasets for learning with graphs. In: *ICML 2020 Workshop on Graph Representation Learning and Beyond (GRL+ 2020)*. (2020). Available online at: [www.graphlearning.io](http://www.graphlearning.io)

**Conflict of Interest:** The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2021 Yim and Leygonie. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.