Geometric Scattering for Graph Data Analysis

Feng Gao [1 2]   Guy Wolf [3]   Matthew Hirn [1 4]

Abstract

We explore the generalization of scattering transforms from traditional (e.g., image or audio) signals to graph data, analogous to the generalization of ConvNets in geometric deep learning, and the utility of extracted graph features in graph data analysis. In particular, we focus on the capacity of these features to retain informative variability and relations in the data (e.g., between individual graphs, or in aggregate), while relating our construction to previous theoretical results that establish the stability of similar transforms to families of graph deformations. We demonstrate the application of our geometric scattering features in graph classification of social network data, and in data exploration of biochemistry data.

1. Introduction

Over the past decade, numerous examples have established that deep neural networks (i.e., cascades of linear operations and simple nonlinearities) typically outperform traditional "shallow" models in various modern machine learning applications, especially given the increasing Big Data availability nowadays. Perhaps the most well known example of the advantages of deep networks is in computer vision, where the utilization of 2D convolutions enable network designs that learn cascades of convolutional filters, which have several advantages over fully connected network architectures, both computationally and conceptually. Indeed, in terms of supervised learning, convolutional neural networks (ConvNets) hold the current state of the art in image classification, and have become the standard machine learning approach to-

---
[1]Department of Computational Math., Science and Engineering, Michigan State University, East Lansing, MI, USA; [2]Department of Plant, Soil & Microbial Sciences, Michigan State University, East Lansing, MI, USA; [3]Department of Mathematics and Statistics, Université de Montréal, Montreal, QC, Canada; [4]Department of Mathematics, Michigan State University, East Lansing, MI, USA. Correspondence to: Guy Wolf <guy.wolf@umontreal.ca>, Matthew Hirn <mhirn@msu.edu>.

wards processing big structured-signal data, including audio and video processing. See, e.g., Goodfellow et al. (2016, Chapter 9) for a detailed discussion.

Beyond their performances when applied to specific tasks, pretrained ConvNet layers have been explored as image feature extractors by freezing the first few pretrained convolutional layers and then retraining only the last few layers for specific datasets or applications (e.g., Yosinski et al., 2014; Oquab et al., 2014). Such transfer learning approaches provide evidence that suitably constructed deep filter banks should be able to extract task-agnostic semantic information from structured data, and in some sense mimic the operation of human visual and auditory cortices, thus supporting the neural terminology in deep learning. An alternative approach towards such universal feature extraction was presented in Mallat (2012), where a deep filter bank, known as the scattering transform, is *designed*, rather than trained, based on predetermined families of distruptive patterns that should be eliminated to extract informative representations. The scattering transform is constructed as a cascade of linear wavelet transforms and nonlinear complex modulus operations that provides features with guaranteed invariance to a predetermined Lie group of operations such as rotations, translations, or scaling. Further, it also provides Lipschitz stability to small diffeomorphisms of the inputted signal.

Following recent interest in geometric deep learning approaches for processing graph-structured data (see, for example, Bronstein et al. (2017) and references therein), several attempts have been made to generalize the scattering transform to graphs (Zou & Lerman, 2018; Gama et al., 2019) and manifolds (Perlmutter et al., 2018), which we will generally term "geometric scattering." These works mostly focus on following the footsteps of Mallat (2012) in establishing the stability of their respective constructions to deformations of input signals or graphs. Their results essentially characterize the type of disruptive information eliminated by geometric scattering, by providing upper bounds for distances between scattering features, phrased in terms of a deformation size. Here, we further explore the notion of geometric scattering features by considering the complimentary question of how much information is retained by them, since stability alone does not ensure useful features in practice (e.g., a constant all-zero map would be stable to any deformation, but would clearly be useless). In other

words, we examine whether a geometric scattering construction, defined and discussed in Sec. 3, can be used as an effective task-independent feature extractor from graphs, and whether the resulting representations provided by them are sufficiently rich to enable intelligible data analysis by applying traditional (Euclidean) methods.

We note that for Euclidean scattering, while stability is established with rigorous theoretical results, the capacity of scattering features to form an effective data representation in practice has mostly been established via extensive empirical examination. Indeed, scattering features have been shown effective in several audio (e.g., Bruna & Mallat, 2013a; Andén & Mallat, 2014; Lostanlen & Mallat, 2015; Andén et al., 2018) and image (e.g., Bruna & Mallat, 2013b; Sifre & Mallat, 2014; Oyallon & Mallat, 2015; Angles & Mallat, 2018) processing applications, and their advantages over learned features are especially relevant in applications with relatively low data availability, such as quantum chemistry and materials science (e.g., Hirn et al., 2017; Eickenberg et al., 2017; 2018; Brumwell et al., 2018).

Similarly, our examination of geometric scattering capacity focuses on empirical results on several data analysis tasks, and on two commonly used graph data types. Our results in Sec. 4.1 show that on social network data, geometric scattering features enable classic RBF-kernel SVM to match, if not outperform, leading graph kernel methods as well as most geometric deep learning ones. These experiments are augmented by additional results in Sec. 4.2 that show the geometric scattering SVM classification rate degrades only slightly when trained on far fewer graphs than is traditionally used in graph classification tasks. On biochemistry data, where graphs represent molecular structures of compounds (e.g., Enzymes or proteins), we show in Sec. 4.3 that scattering features enable significant dimensionality reduction. Finally, to establish their descriptive qualities, in Sec. 4.4 we use geometric scattering features extracted from enzyme data (Borgwardt et al., 2005a) to infer emergent patterns of enzyme commission (EC) exchange preferences in enzyme evolution, validated with established knowledge from Cuesta et al. (2015). Taken together, these results illustrate the power of the geometric scattering approach as both a relevant mathematical model for geometric deep learning, and as a suitable tool for modern graph data analysis.

2. Graph Random Walks and Graph Wavelets

The Euclidean scattering transform is constructed using wavelets defined on $\mathbb{R}^d$. In order to extend this construction to graphs, we define graph wavelets as the difference between lazy random walks that have propagated at different time scales, which mimics classical wavelet constructions found in Meyer (1993) and more recent constructions found in Coifman & Maggioni (2006). The underpinnings for this

construction arise out of graph signal processing, and in particular the properties of the graph Laplacian.

Let $G = (V, E, W)$ be a weighted graph, consisting of n vertices $V = \{v_1, \ldots, v_n\}$, edges $E \subseteq \{(v_\ell, v_m) : 1 \leq \ell, m \leq n\}$, and weights $W = \{w(v_\ell, v_m) > 0 : (v_\ell, v_m) \in E\}$. Note that unweighted graphs are considered as a special case, by setting $w(v_\ell, v_m) = 1$ for each $(v_\ell, v_m) \in E$. Define the $n \times n$ (weighted) adjacency matrix $\mathbf{A}_G = \mathbf{A}$ of G by $\mathbf{A}(v_\ell, v_m) = w(v_\ell, v_m)$ if $(v_\ell, v_m) \in E$ and zero otherwise, where we use the notation $\mathbf{A}(v_\ell, v_m)$ to denote the (ℓ, m) entry of the matrix $\mathbf{A}$ so as to emphasize the correspondence with the vertices in the graph and to reserve sub-indices for enumerating objects. Define the (weighted) degree of vertex v_ℓ as $\deg(v_\ell) = \sum_m \mathbf{A}(v_\ell, v_m)$ and the corresponding diagonal $n \times n$ degree matrix $\mathbf{D}$ given by $\mathbf{D}(v_\ell, v_\ell) = \deg(v_\ell)$, $\mathbf{D}(v_\ell, v_m) = 0$, $\ell \neq m$. Finally, the $n \times n$ graph Laplacian matrix $\mathbf{L}_G = \mathbf{L}$ on G is defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$, and its normalized version is $\mathbf{N} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$. We focus on the latter due to its close relationship with graph random walks.

The normalized graph Laplacian is a symmetric, real valued positive semi-definite matrix, and thus has n non-negative eigenvalues. Furthermore, if we set $\mathbf{0} = (0, \ldots, 0)^T$ to be the $n \times 1$ vector of all zeroes, and $\mathbf{d}(v_\ell) = \deg(v_\ell)$ to be the $n \times 1$ degree vector, then one has $\mathbf{N}\mathbf{d}^{1/2} = \mathbf{0}$ (where the square root is understood to be taken entrywise). Therefore 0 is an eigenvalue of $\mathbf{N}$ and we write the n eigenvalues of $\mathbf{N}$ as $0 = \lambda_0 \leq \lambda_1 \leq \cdots \leq \lambda_{n-1} \leq 2$ with corresponding $n \times 1$ orthonormal eigenvectors $\boldsymbol{\varphi}_0, \boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_{n-1}$. If the graph G is connected, then $\lambda_1 > 0$. In order to simplify the following discussion we assume that this is the case, although the discussion below can be amended to include disconnected graphs as well.

One can show $\boldsymbol{\varphi}_0 = \mathbf{d}^{1/2}/\|\mathbf{d}^{1/2}\|$, meaning $\boldsymbol{\varphi}_0$ is non-negative. Since every other eigenvector is orthogonal to $\boldsymbol{\varphi}_0$ (and thus must take positive and negative values), it is natural to view the eigenvectors $\boldsymbol{\varphi}_k$ as the Fourier modes of the graph G, with a frequency magnitude proportional to λ_k. The fact that $\boldsymbol{\varphi}_0$ is in general non-constant, as opposed to the zero frequency mode on the torus or real line, reflects the non-uniform distribution of vertices in non-regular graphs. Let $\mathbf{x} : V \to \mathbb{R}$ be a signal defined on the vertices of the graph G, which we will consider as an $n \times 1$ vector with entries $\mathbf{x}(v_\ell)$. It follows that the Fourier transform of $\mathbf{x}$ can be defined as $\hat{\mathbf{x}}(k) = \mathbf{x} \cdot \boldsymbol{\varphi}_k$, where $\mathbf{x} \cdot \mathbf{y}$ is the standard dot product. This analogy is one of the foundations of graph signal processing and indeed we could use this correspondence to define wavelet operators on the graph G, as in Hammond et al. (2011). Rather than follow this path, though, we instead take a related path similar to Coifman & Maggioni (2006) and Gama et al. (2019) by defining the graph wavelet operators in terms of random walks defined

on G, which will avoid diagonalizing $\mathbf{N}$ and will allow us to control the "spatial" graph support of the filters directly.

Define the $n \times n$ lazy random walk matrix as $\mathbf{P} = \frac{1}{2}\left(\mathbf{I} + \mathbf{AD}^{-1}\right)$. Note that the column sums of $\mathbf{P}$ are all one. It follows that $\mathbf{P}$ acts as a Markov operator, mapping probability distributions to probability distribution. We refer to $\mathbf{P}$ as a lazy random walk matrix since $\mathbf{P}^t$ governs the probability distribution of a lazy random walk after t steps. A single realization of a random walk is a walk (in the graph theoretic sense) $v_{\ell_0}, v_{\ell_1}, v_{\ell_2}, \ldots$ in which the steps are chosen randomly; lazy random walks allow for $v_{\ell_i} = v_{\ell_{i+1}}$. More precisely, suppose that $\boldsymbol{\mu}_0(v_\ell) \geq 0$ for each vertex v_ℓ and $\|\boldsymbol{\mu}_0\|_1 = 1$, so that $\boldsymbol{\mu}_0$ is a probability distribution on G. We take $\boldsymbol{\mu}_0(v_\ell)$ as the probability of a random walk starting at vertex $v_{\ell_0} = v_\ell$. One can verify that $\boldsymbol{\mu}_1 = \mathbf{P}\boldsymbol{\mu}_0$ is also a probability distribution; each entry $\boldsymbol{\mu}_1(v_\ell)$ gives the probability of the random walk being located at $v_{\ell_1} = v_\ell$ after one step. The probability distribution for the location of the random walk after t steps is $\boldsymbol{\mu}_t = \mathbf{P}^t\boldsymbol{\mu}_0$.

The operator $\mathbf{P}$ can be considered a low pass operator, meaning that $\mathbf{Px}$ replaces $\mathbf{x}(v_\ell)$ with localized averages of $\mathbf{x}(v_\ell)$ for any $\mathbf{x}$. Indeed, expanding out $\mathbf{Px}(v_\ell)$ one observes that $\mathbf{Px}(v_\ell)$ is the weighted average of $\mathbf{x}(v_\ell)$ and the values $\mathbf{x}(v_m)$ for the neighbors v_m of v_ℓ. Similarly, the value $\mathbf{P}^t\mathbf{x}(v_\ell)$ is the weighted average of $\mathbf{x}(v_\ell)$ with all values $\mathbf{x}(v_m)$ such that v_m is within t steps of v_ℓ.

Low pass operators defined on Euclidean space retain the low frequencies of a function while suppressing the high frequencies. The random walk matrix $\mathbf{P}$ behaves similarly. Indeed, $\mathbf{P}$ is diagonalizable with n eigenvectors $\boldsymbol{\phi}_k = \mathbf{D}^{1/2}\boldsymbol{\varphi}_k$ and eigenvalues $\omega_k = 1 - \lambda_k/2$. Let $\mathbf{y_x} = \mathbf{D}^{-1/2}\mathbf{x}$ be a density normalized version of $\mathbf{x}$ and set $\mathbf{x}_t = \mathbf{P}^t\mathbf{x}$; then one can show

$$\mathbf{y}_{\mathbf{x}_t} = \widehat{\mathbf{y_x}}(0)\boldsymbol{\varphi}_0 + \sum_{k=1}^{n-1} \omega_k^t \widehat{\mathbf{y_x}}(k)\boldsymbol{\varphi}_k. \qquad (1)$$

Thus, since $0 \leq \omega_k < 1$ for $k \geq 1$, the operator $\mathbf{P}^t$ preserves the zero frequency of $\mathbf{x}$ while suppressing the high frequencies, up to a density normalization.

High frequency responses of $\mathbf{x}$ can be recovered in multiple different fashions, but we utilize multiscale wavelet transforms that group the non-zero frequencies of G into approximately dyadic bands. As shown in Mallat (2012, Lemma 2.12), wavelet transforms are provably stable operators in the Euclidean domain, and the proof of Zou & Lerman (2018, Theorem 5.1) indicates that similar results on graphs may be possible. Furthermore, the multiscale nature of wavelet transforms will allow the resulting geometric scattering transform (Sec. 3) to traverse the entire graph G in one layer, which is valuable for obtaining global descriptions of G. Following Coifman & Maggioni (2006),
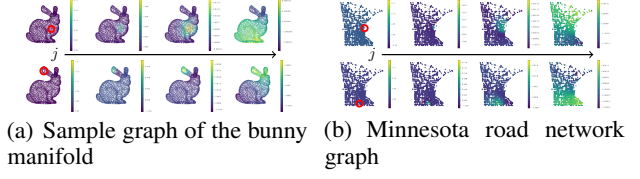

(a) Sample graph of the bunny manifold

(b) Minnesota road network graph

Figure 1. Wavelets $\boldsymbol{\Psi}_j$ for increasing scale 2^j left to right, applied to Diracs centered at two different locations (marked by red circles) in two different graphs. Vertex colors indicate wavelet values (corresponding to colorbars for each plot), ranging from yellow/green indicating positive values to blue indicating negative values. Both graphs are freely available from PyGSP (2018).

define the $n \times n$ wavelet matrix at the scale 2^j as

$$\boldsymbol{\Psi}_j = \mathbf{P}^{2^{j-1}} - \mathbf{P}^{2^j} = \mathbf{P}^{2^{j-1}}(\mathbf{I} - \mathbf{P}^{2^{j-1}}). \qquad (2)$$

A similar calculation as the one required for (1) shows that $\boldsymbol{\Psi}_j\mathbf{x}$ partially recovers $\widehat{\mathbf{y_x}}(k)$ for $k \geq 1$. The value $\boldsymbol{\Psi}_j\mathbf{x}(v_\ell)$ aggregates the signal information $\mathbf{x}(v_m)$ from the vertices v_m that are within 2^j steps of v_ℓ, but does not average the information like the operator $\mathbf{P}^{2^j}$. Instead, it responds to sharp transitions or oscillations of the signal $\mathbf{x}$ within the neighborhood of v_ℓ with radius 2^j (in terms of the graph path distance). The smaller the wavelet scale 2^j, the higher the frequencies $\boldsymbol{\Psi}_j\mathbf{x}$ recovers in $\mathbf{x}$. The wavelet coefficients up to the scale 2^J are:

$$\boldsymbol{\Psi}^{(J)}\mathbf{x}(v_\ell) = [\boldsymbol{\Psi}_j\mathbf{x}(v_\ell) : 1 \leq j \leq J]. \qquad (3)$$

Figure 1 plots the wavelets on two different graphs.

3. Geometric Scattering on Graphs

A geometric wavelet scattering transform follows a similar construction as the (Euclidean) wavelet scattering transform of Mallat (2012), but leverages a graph wavelet transform. In this paper we utilize the wavelet transform defined in (3) of the previous section, but remark that in principle any graph wavelet transform could be used (see, e.g., Zou & Lerman, 2018). In Sec. 3.1 we define the graph scattering transform, in Sec. 3.2 we discuss its relation to other recently proposed graph scattering constructions (Gama et al., 2019; Zou & Lerman, 2018), and in Sec. 3.3 we describe several of its desirable properties as compared to other geometric deep learning algorithms on graphs.

3.1. Geometric scattering definitions

Machine learning algorithms that compare and classify graphs must be invariant to graph isomorphism, i.e., re-indexations of the vertices and corresponding edges. A common way to obtain invariant graph features is via summation operators, which act on a signal $\mathbf{x} = \mathbf{x}_G$ that can be defined on any graph G, e.g., $\mathbf{x}(v_\ell) = \deg(v_\ell)$. The

geometric scattering transform, which is described in the remainder of this section, follows such an approach.

The simplest summation operator computes the sum of the responses of the signal $\mathbf{x}$. As described in Verma & Zhang (2018), this invariant can be complemented by higher order summary statistics of $\mathbf{x}$, the collection of which are statistical moments, and which are also referred to as "capsules" in that work. For example, the unnormalized q^{th} moments of $\mathbf{x}$ yield the following "zero" order scattering moments:

$$Sx(q) = \sum_{\ell=1}^{n} \mathbf{x}(v_\ell)^q, \quad 1 \le q \le Q \tag{4}$$

We can also replace (4) with normalized (i.e., standardized) moments of $\mathbf{x}$, in which case we store its mean ($q = 1$), variance ($q = 2$), skew ($q = 3$), kurtosis ($q = 4$), and so on. In what follows we discuss the unnormalized moments since their presentation is simpler. The invariants $Sx(q)$ do not capture the full variability of $\mathbf{x}$ and hence the graph G upon which the signal $\mathbf{x}$ is defined. We thus complement these moments with summary statistics derived from the wavelet coefficients of $\mathbf{x}$, which will lead naturally to the graph ConvNet structure of the geometric scattering transform.

Observe, analogously to the Euclidean setting, that in computing $Sx(1)$, which is the summation of $\mathbf{x}(v_\ell)$ over V, we have captured the zero frequency of $\mathbf{y_x} = \mathbf{D}^{-1/2}\mathbf{x}$ since $\sum_{\ell=1}^{n} \mathbf{x}(v_\ell) = \mathbf{x} \cdot \mathbf{1} = \mathbf{y_x} \cdot \mathbf{d}^{1/2} = \|\mathbf{d}^{1/2}\|\widehat{\mathbf{y_x}}(0)$. Higher order moments of $\mathbf{x}$ can incorporate the full range of frequencies in $\mathbf{x}$, e.g. $Sx(2) = \sum_{\ell=1}^{n} \mathbf{x}(v_\ell)^2 = \sum_{k=1}^{n} \widehat{\mathbf{x}}(k)^2$, but they are mixed into one invariant coefficient. We can separate and recapture the high frequencies of $\mathbf{x}$ by computing its wavelet coefficients $\mathbf{\Psi}^{(J)}\mathbf{x}$, which were defined in (3). However, $\mathbf{\Psi}^{(J)}\mathbf{x}$ is not invariant to permutations of the vertex indices; in fact, it is equivariant. Before summing the individual wavelet coefficient vectors $\mathbf{\Psi}_j\mathbf{x}$, though, we must first apply a pointwise nonlinearity. Indeed, define $\mathbf{1} = (1, \dots, 1)^T$ to be the $n \times 1$ vector of all ones, and note that $\mathbf{P}^T\mathbf{1} = \mathbf{1}$, meaning that $\mathbf{1}$ is a left eigenvector of $\mathbf{P}$ with eigenvalue 1. It follows that $\mathbf{\Psi}_j^T\mathbf{1} = \mathbf{0}$ and thus $\sum_{\ell=1}^{n} \mathbf{\Psi}_j\mathbf{x}(v_\ell) = \mathbf{\Psi}_j\mathbf{x} \cdot \mathbf{1} = \mathbf{1}^T\mathbf{\Psi}_j\mathbf{x} = 0$.

We thus apply the absolute value nonlinearity, to obtain nonlinear equivariant coefficients $|\mathbf{\Psi}^{(J)}\mathbf{x}| = \{|\mathbf{\Psi}_j\mathbf{x}| : 1 \le j \le J\}$. We use absolute value because it is equivariant to vertex permutations, non-expansive, and when combined with traditional wavelet transforms on Euclidean domains, yields a provably stable scattering transform for $q = 1$. Furthermore, initial theoretical results in Zou & Lerman (2018) and Gama et al. (2019) indicate that similar graph based scattering transforms possess certain types of stability properties as well. As in (4), we extract invariant coefficients from $|\mathbf{\Psi}_j\mathbf{x}|$ by computing its moments, which define the first order geometric scattering moments:

$$Sx(j, q) = \sum_{\ell=1}^{n} |\mathbf{\Psi}_j\mathbf{x}(v_\ell)|^q, \ 1 \le j \le J, \ 1 \le q \le Q \tag{5}$$

These first order scattering moments aggregate complimentary multiscale geometric descriptions of G into a collection of invariant multiscale statistics. These invariants give a finer partition of the frequency responses of $\mathbf{x}$. For example, whereas $Sx(2)$ mixed all frequencies of $\mathbf{x}$, we see that $Sx(j, 2)$ only mixes the frequencies of $\mathbf{x}$ captured by $\mathbf{\Psi}_j$.

First order geometric scattering moments can be augmented with second order geometric scattering moments by iterating the graph wavelet and absolute value transforms. These moments are defined as:

$$Sx(j, j', q) = \sum_{\ell=1}^{n} ||\mathbf{\Psi}_{j'}|\mathbf{\Psi}_j\mathbf{x}(v_\ell)||^q, \ \begin{matrix} 1 \le j < j' \le J \\ 1 \le q \le Q \end{matrix}, \tag{6}$$

which consists of reapplying the wavelet transform operator $\mathbf{\Psi}^{(J)}$ to each $|\mathbf{\Psi}_j\mathbf{x}|$ and computing the summary statistics of the magnitudes of the resulting coefficients. The intermediate equivariant coefficients $|\mathbf{\Psi}_{j'}|\mathbf{\Psi}_j\mathbf{x}||$ and resulting invariant statistics $Sx(j, j', q)$ couple two scales 2^j and $2^{j'}$ within the graph G, creating features that bind patterns of smaller subgraphs within G with patterns of larger subgraphs (e.g., circles of friends of individual people with larger community structures in social network graphs). The transform can be iterated additional times, leading to third order features and beyond, and thus has the general structure of a graph ConvNet.

The collection of graph scattering moments $Sx = \{Sx(q), \ Sx(j, q), \ Sx(j, j', q)\}$ (illustrated in Fig. 2(a)) provides a rich set of multiscale invariants of the graph G. These can be used in supervised settings as input to graph classification or regression models, or in unsupervised settings to embed graphs into a Euclidean feature space for further exploration, as demonstrated in Sec. 4.

3.2. Stability and capacity of geometric scattering

In order to assess the utility of scattering features for representing graphs, two properties have to be considered: stability and capacity. First, the stability property aims to provide an upper bound on distances between similar graphs that only differ by types of deformations that can be treated as noise. This property has been the focus of both Zou & Lerman (2018) and Gama et al. (2019), and in particular the latter shows that a diffusion scattering transform yields features that are stable to graph structure deformations whose size can be computed via the diffusion framework (Coifman & Maggioni, 2006) that forms the basis for their construction. While there are some technical differences between the geometric scattering here and the diffusion scattering
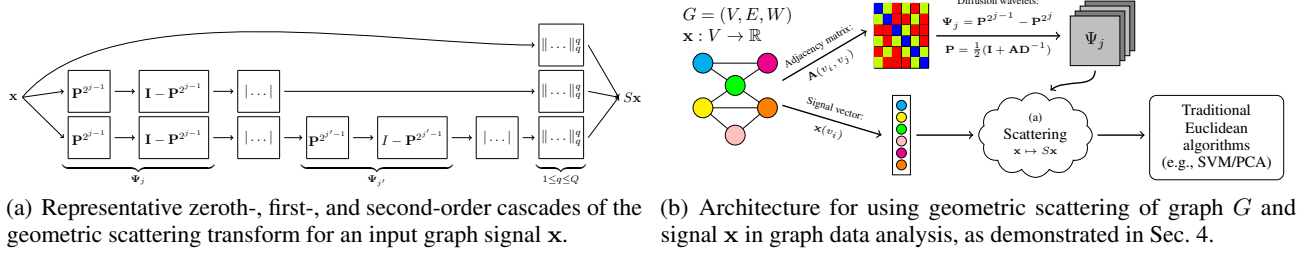
(a) Representative zeroth-, first-, and second-order cascades of the geometric scattering transform for an input graph signal $\mathbf{x}$.

(b) Architecture for using geometric scattering of graph G and signal $\mathbf{x}$ in graph data analysis, as demonstrated in Sec. 4.

Figure 2. Illustration of (a) the proposed scattering feature extraction (see eqs. 4, 5, and 6), and (b) its application for graph data analysis.

in Gama et al. (2019), these constructions are sufficiently similar that we can expect both of them to have analogous stability properties. Therefore, we mainly focus here on the complementary property of the scattering transform capacity to provide a rich feature space for representing graph data without eliminating informative variance in them.

We note that even in the classical Euclidean case, while the stability of scattering transforms to deformations can be established analytically (Mallat, 2012), their capacity is typically examined by empirical evidence when applied to machine learning tasks (e.g., Bruna & Mallat, 2011; Sifre & Mallat, 2012; Andén & Mallat, 2014). Similarly, in the graph processing settings, we examine the capacity of our proposed geometric scattering features via their discriminative power in graph data analysis tasks, which are described in detail in Sec. 4. We show that geometric scattering enables graph embedding in a relatively low dimensional Euclidean space, while preserving insightful properties in the data. Beyond establishing the capacity of our specific construction, these results also indicate the viability of graph scattering transforms as universal feature extractors on graph data, and complement the stability results established in Zou & Lerman (2018) and Gama et al. (2019).

3.3. Geometric scattering compared to other feed forward graph ConvNets

We give a brief comparison of geometric scattering with other graph ConvNets, with particular interest in isolating the key principles for building accurate graph ConvNet classifiers. Like several other successful graph neural networks, the graph scattering transform is equivariant to vertex permutations (i.e., commutes with them) until the final features are extracted. This idea has been discussed in depth in various articles, including Kondor et al. (2018), so we limit the discussion to observing that the geometric scattering transform thus propagates nearly all of the information in $\mathbf{x}$ through the multiple wavelet and absolute value layers, since only the absolute value operation removes information on $\mathbf{x}$. As in Verma & Zhang (2018), we aggregate covariant responses via multiple summary statistics (i.e., moments), which are referred to there as a capsule. In the scattering context, at least, this idea is in fact not new and has been

previously used in the Euclidean setting for the regression of quantum mechanical energies in Eickenberg et al. (2018; 2017) and texture synthesis in Bruna & Mallat (2018). However, unlike many deep learning classifiers (graph included), a graph scattering transform extracts invariant statistics at each layer/order. These intermediate layer statistics, while necessarily losing some information in $\mathbf{x}$ (and hence G), provide important coarse geometric invariants that eliminate needless complexity in subsequent classification or regression. Furthermore, such layer by layer statistics have proven useful in characterizing signals of other types (e.g., texture synthesis in Gatys et al., 2015).

A graph wavelet transform $\mathbf{\Psi}^{(J)}\mathbf{x}$ decomposes the geometry of G through the lens of $\mathbf{x}$, along different scales. Graph ConvNet algorithms also obtain multiscale representations of G, but several works, including Atwood & Towsley (2016) and Zhang et al. (2018), propagate information via a random walk. While random walk operators like $\mathbf{P}^t$ act at different scales on the graph G, per the analysis in Sec. 2 we see that $\mathbf{P}^t$ for any t will be dominated by the low frequency responses of $\mathbf{x}$. While subsequent nonlinearities may be able to recover this high frequency information, the resulting transform will most likely be unstable due to the suppression and then attempted recovery of the high frequency content of $\mathbf{x}$. Alternatively, features derived from $\mathbf{P}^t\mathbf{x}$ may lose the high frequency responses of $\mathbf{x}$, which are useful in distinguishing similar graphs. The graph wavelet coefficients $\mathbf{\Psi}^{(J)}\mathbf{x}$, on the other hand, respond most strongly within bands of nearly non-overlapping frequencies, each with a center frequency k_j that depends on $\mathbf{\Psi}_j$.

Finally, graph labels are often complex functions of both local and global subgraph structure within G. While graph ConvNets are adept at learning local structure within G, as detailed in Verma & Zhang (2018) they require many layers to obtain features that aggregate macroscopic patterns in the graph. This is due to the use of fixed size filters, which often only incorporate information from the neighbors of a vertex. The training of such networks is difficult due to the limited size of many graph classification databases (see the supplementary information). Geometric scattering transforms have two advantages in this regard: (a) the wavelet filters are designed; and (b) they are multiscale, thus incorporating

macroscopic graph patterns in every layer/order.

4. Application & Results

To establish the geometric scattering features as an effective graph representation for data analysis, we examine their performance here in four graph data analysis applications. Namely, in Sec. 4.1 we consider graph classification on social networks (from Yanardag & Vishwanathan, 2015), in Sec. 4.2 we consider the impact of low training data availability on classification, in Sec. 4.3 we examine dimensionality reduction aspects of geometric scattering, and finally, in Sec. 4.4 we consider data exploration of enzyme graphs, where geometric scattering enables unsupervised (descriptive) recovery of EC change preferences in enzyme evolution. A common theme in all these applications is the application of geometric scattering as an unsupervised task-independent feature extraction that embeds input graphs of varying sizes (with associated graph signals) into a Euclidean space formed by scattering features. Then, the extracted feature vectors are passed to traditional (Euclidean) machine learning algorithms, such as SVM for classification or PCA for dimensionality reduction, to perform downstream analysis. Our results show that our scattering features provide simplified representation (e.g., in dimensionality and extrapolation ability) of input graphs, which we conjecture is a result of their stability properties, while also being sufficiently rich to capture meaningful relations between graphs for predictive and descriptive purposes.

4.1. Graph classification on social networks

As a first application of geometric scattering, we apply it to graph classification of social network data taken from Yanardag & Vishwanathan (2015). In particular, this work introduced six social network data sets extracted from scientific collaborations (COLLAB), movie collaborations (IMDB-B & IMDB-M), and Reddit discussion threads (REDDIT-B, REDDIT-5K, REDDIT-12K). There are also biochemistry data sets often used in the graph classification literature; for completeness, we include in the supplemental materials further results on these data sets. A brief description of each data set can also be found in the supplement.

The social network data provided by Yanardag & Vishwanathan (2015) contains graph structures but no associated graph signals. Therefore we compute the eccentricity (for connected graphs) and clustering coefficient of each vertex, and use these as input signals to the geometric scattering transform. In principle, any general node characteristic could be used, although we remark that $\mathbf{x} = \mathbf{d}$, the vertex degree vector, is not useful in our construction since $\mathbf{\Psi}_j \mathbf{d} = \mathbf{0}$. After computing the scattering moments[1] of

these two input signals, they are concatenated to form a single vector. This scattering feature vector is a consistent Euclidean representation of the graph, which is independent of the original graph sizes (i.e., number of vertices or edges), and thus we can apply any traditional classifier to it. In particular, we use here the standard SVM classifier with an RBF kernel, which is popular and effective in many applications and also performs well in this case.

We evaluate the classification results of our SVM-based geometric scattering classification (GS-SVM) using ten-fold cross validation (explained in the supplement), which is standard practice in other graph classification works. We compare our results to 10 prominent methods that report results for most, if not all, of the considered datasets. Out of these, four are graph kernel methods: Weisfeiler-Lehman graph kernels (WL, Shervashidze et al., 2011), Graphlet kernels (Shervashidze et al., 2009), deep graph kernels (DGK, Yanardag & Vishwanathan, 2015), and Weisfeiler-Lehman optimal assignment kernels (WL-OA, Kriege et al., 2016). The other six are recent geometric deep learning algorithms: deep graph convolutional neural network (DGCNN, Zhang et al., 2018), 2D convolutional neural networks (2DCNN, Tixier et al., 2017), Patchy-san (PSCN, Niepert et al., 2016, with $k = 10$), graph capsule convolutional neural networks (GCAPS-CNN, Verma & Zhang, 2018), recurrent neural network autoencoders (S2S-N2N-PP, Taheri et al., 2018), and the graph isomorphism network (GIN, Xu et al., 2019).

Following the standard format of reported classification performances for these methods (per their respective references, see also the supplement), our results are reported in the form of average accuracy $\pm$ standard deviation (in percentages) over the ten cross-validation folds. We note that since some methods are not reported for all datasets, we mark N/A when appropriate. Table 1 reports the results.

The geometric scattering transform and related variants presented in Zou & Lerman (2018) and Gama et al. (2019) is a mathematical model for graph ConvNets. However, it is natural to ask if this model accurately reflects what is done in practice. A useful model may not obtain state of the art performance, but should be competitive with the current state of the art, lest the model may not capture the underlying complexity of the most powerful methods. Examining Table 1 one can see that the GS-SVM classifier matches or outperforms all but the two most recent methods, i.e., S2S-N2N-PP (Taheri et al., 2018) and GIN (Xu et al., 2019). With regards to these two approaches, the GS-SVM outperforms S2S-N2N-PP (Taheri et al., 2018) on $3/6$ datasets. Finally, while GIN (Xu et al., 2019) outperforms geometric scattering on $5/6$ datasets, the results on

---

[1]We use the normalized scattering moments for classification, since they perform slightly better than the un-normalized moments. Also we use $J = 5$ and $q = 4$ for all scattering feature generations.

Table 1. Comparison of the proposed GS-SVM classifier with leading graph kernel and deep learning methods on social graph datasets.

	COLLAB	IMDB-B	IMDB-M	REDDIT-B	REDDIT-5K	REDDIT-12K	
WL	77.82 ± 1.45	71.60 ± 5.16	N/A	78.52 ± 2.01	50.77 ± 2.02	34.57 ± 1.32	Graph kernel
Graphlet	73.42 ± 2.43	65.40 ± 5.95	N/A	77.26 ± 2.34	39.75 ± 1.36	25.98 ± 1.29	
WL-OA	80.70 ± 0.10	N/A	N/A	89.30 ± 0.30	N/A	N/A	
DGK	73.00 ± 0.20	66.90 ± 0.50	44.50 ± 0.50	78.00 ± 0.30	41.20 ± 0.10	32.20 ± 0.10	
DGCNN	73.76 ± 0.49	70.03 ± 0.86	47.83 ± 0.85	N/A	48.70 ± 4.54	N/A	Deep learning
2D CNN	71.33 ± 1.96	70.40 ± 3.85	N/A	89.12 ± 1.70	52.21 ± 2.44	48.13 ± 1.47	
PSCN ($k = 10$)	72.60 ± 2.15	71.00 ± 2.29	45.23 ± 2.84	86.30 ± 1.58	49.10 ± 0.70	41.32 ± 0.42	
GCAPS-CNN	77.71 ± 2.51	71.69 ± 3.40	48.50 ± 4.10	87.61 ± 2.51	50.10 ± 1.72	N/A	
S2S-P2P-NN	81.75 ± 0.80	73.80 ± 0.70	51.19 ± 0.50	86.50 ± 0.80	52.28 ± 0.50	42.47 ± 0.10	
GIN-0 (MLP-SUM)	80.20 ± 1.90	75.10 ± 5.10	52.30 ± 2.80	92.40 ± 2.50	57.50 ± 1.50	N/A	
GS-SVM	79.94 ± 1.61	71.20 ± 3.25	48.73 ± 2.32	89.65 ± 1.94	53.33 ± 1.37	45.23 ± 1.25	

COLLAB and IMDB-B are not statistically significant, and on the REDDIT datasets the geometric scattering approach trails only GIN (Xu et al., 2019). We thus conclude that the geometric scattering transform yields a rich set of invariant statistical moments, which have nearly the same capacity as the current state of the art in graph neural networks.

4.2. Classification with low training-data availability

Many modern deep learning methods require large amounts of training data to generate representative features. On the contrary, geometric scattering features are based on each graph without any training processes. In this section, we demonstrate the performance of the GS-SVM under low training-data availability and show that the scattering features can embed enough graph information that even under extreme conditions (e.g. only 20% training data), they can still maintain relatively good classification results.

We performed graph classification under four training/validation/test splits: 80%/10%/10%, 70%/10%/20%, 40%/10%/50% and 20%/10%/70%. We did 10-fold, 5-fold and 2-fold cross validation for the first three splits. For the last split, we randomly formed a 10 folds pool, from which we randomly selected 3 folds for training/validation and repeated this process ten times. Detailed classification results can be found in the supplement. Following Sec. 4.1,

we discuss the classification accuracy on six social datasets under these splits. When the training data is reduced from 90% to 80%, the classification accuracy in fact increased by 0.047%, which shows the GS-SVM classification accuracy is not affected by the decrease in training size. Further reducing the training size to 50% results in an average decrease of classification accuracy of 1.40% while from 90% to 20% causes an average decrease of 3.00%. Fig. 3 gives a more nuanced statistical description of these results.

4.3. Dimensionality reduction

We now consider the viability of scattering-based embedding for dimensionality reduction of graph data. As a representative example, we consider here the ENZYMES dataset introduced in Borgwardt et al. (2005b), which contains 600 enzymes evenly split into six enzyme classes (i.e., 100 enzymes from each class). While the Euclidean notion of dimensionality is not naturally available in graph data, we note that graphs in this dataset have, on average, 124.2 edges, 29.8 vertices, and 3 features per vertex. Therefore, the data here can be considered significantly high dimensional in its original representation, which is not amenable to traditional dimensionality reduction techniques.

To perform scattering-based dimensionality reduction, we applied PCA to geometric scattering features extracted from input enzyme graphs in the data, while choosing the number of principal components to capture 99%, 90%, 80% and 50% explained variance. For each of these thresholds, we computed the mean classification accuracy (with ten-fold cross validation) of SVM applied to the GS-PCA low dimensional space, as well as the dimensionality of this space. The relation between dimensionality, explained variance, and SVM accuracy is shown in Fig. 3, where we can observe that indeed geometric scattering combined with PCA enables significant dimensionality reduction (e.g., to $\mathbb{R}^{16}$ with 90% exp. variance) with only a small impact on classification accuracy. Finally, we also consider the PCA dimensionality of each individual enzyme class in the data (in the scattering feature space), as we expect scattering to reduce the variability in each class w.r.t. the full feature space. Indeed, in this case, individual classes have 90% exp. variance PCA dimensionality ranging between 6 and 10, which is
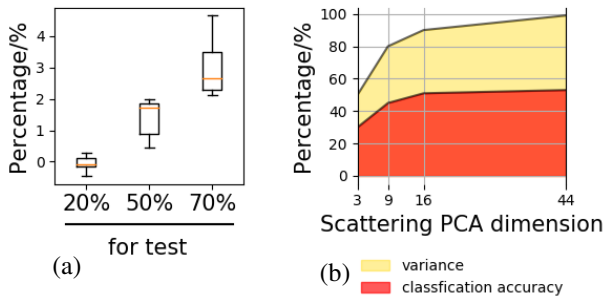

Figure 3. (a) Box plot showing the drop in SVM classification accuracy over social graph datasets when reducing training set size (horizontal axis marks portion of data used for testing); (b) Relation between explained variance, SVM classification accuracy, and PCA dimensions over scattering features in ENZYMES dataset.

significantly lower than the 16 dimensions of the entire PCA space. We note that similar results can also be observed for the social network data discussed in previous sections, where on average 90% explained variances are captured by nine dimensions, yielding a drop of 3.81% in mean SVM accuracy; see the supplement for complete results.

4.4. Data exploration: Enzyme class exchange preferences

Geometric scattering essentially provides a task independent representation of graphs in a Euclidean feature space. Therefore, it is not limited to supervised learning applications, and can be also utilized for exploratory graph-data analysis, as we demonstrate in this section. We focus our discussion in particular on the ENZYMES dataset described in the previous section. Here, geometric scattering features can be considered as providing "signature" vectors for individual enzymes, which can be used to explore interactions between the six top level enzyme classes, labeled by their Enzyme Commission (EC) numbers (Borgwardt et al., 2005a). In order to emphasize the properties of scattering-based feature extraction, rather than downstream processing, we mostly limit our analysis of the scattering feature space to linear operations such as principal component analysis (PCA).

To explore the scattering feature space, and the richness of information captured by it, we use it to infer relations between EC classes. First, for each enzyme e, with scattering feature vector $\mathbf{v}_e$ (i.e., with $S\mathbf{x}$ for all vertex features $\mathbf{x}$), we compute its distance from class EC-j, with PCA subspace $\mathcal{C}_j$, as the projection distance: $\text{dist}(e, \text{EC-}j) = \|\mathbf{v}_e - \text{proj}_{\mathcal{S}_j}\mathbf{v}_e\|$. Then, for each enzyme class EC-i, we compute the mean distance of enzymes in it from the subspace of each EC-j class as $D(i,j) = \text{mean}\{\text{dist}(e, \text{EC-}j) : e \in \text{EC-}i\}$.

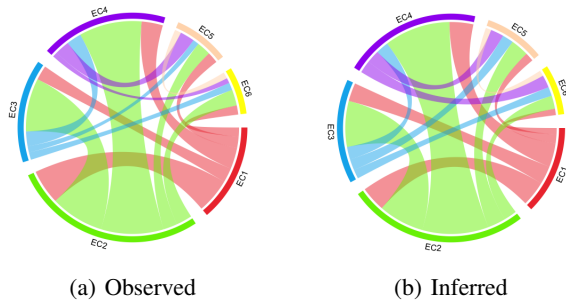(a) Observed   (b) Inferred

Figure 4. Comparison of EC exchange preferences in enzyme evolution: (a) observed in Cuesta et al. (2015), and (b) inferred from scattering features via $\text{pref}(\text{EC-}i, \text{EC-}j) := w_j \cdot \left[\min\left\{\frac{D(i,j)}{D(i,i)}, \frac{D(j,i)}{D(j,j)}\right\}\right]^{-1}$; w_j = portion of enzymes in EC-j that choose another EC as their nearest subspace; $D(i,j)$ = mean dist. of enzymes in EC-i from PCA (90% exp. var.) subspace of EC-j. Our inference (b) mainly recovers (a).

These distances are summarized in the supplement, as well as the proportion of points from each class that have their true EC as their nearest (or second nearest) subspace in the scattering feature space. In general, 48% of enzymes select their true EC as the nearest subspace (with additional 19% as second nearest), but these proportions vary between individual EC classes. Finally, we use these scattering-based distances to infer EC exchange preferences during enzyme evolution, which are presented in Fig. 4 and validated with respect to established preferences observed and reported in Cuesta et al. (2015). We note that the result there is observed independently from the ENZYMES dataset. In particular, the portion of enzymes considered from each EC is different between these data, since Borgwardt et al. (2005b) took special care to ensure each EC class in ENZYMES has exactly 100 enzymes in it. However, we notice that in fact the portion of enzymes (in each EC) that choose the wrong EC as their nearest subspace, which can be considered as EC "incoherence" in the scattering feature space, correlates well with the proportion of evolutionary exchanges generally observed for each EC in Cuesta et al. (2015), and therefore we use these as EC weights (see Fig. 4). Our results in Fig. 4 demonstrate that scattering features are sufficiently rich to capture relations between enzyme classes, and indicate that geometric scattering has the capacity to uncover descriptive and exploratory insights in graph data analysis.

5. Conclusion

We presented the geometric scattering transform as a deep filter bank for feature extraction on graphs, which generalizes the Euclidean scattering transform. A reasonable criticism of the scattering theory approach to understanding geometric deep learning is that it is not clear if the scattering model is a suitable facsimile for powerful graph neural networks that are obtaining impressive results on graph classification tasks and related graph data analysis problems. In this paper we showed that in fact, at least empirically, this line of criticism is unfounded and indeed further theoretical study of geometric scattering transforms on graphs is warranted. Our evaluation results on graph classification and data exploration show the potential of the produced scattering features to serve as universal representations of graphs. Indeed, classification using these features with relatively simple classifier models, dimension reduced feature sets, and small training sets nevertheless reach high accuracy results on most commonly used graph classification datasets. Finally, the geometric scattering features provide a new way for computing and considering global graph representations, independent of specific learning tasks. They raise the possibility of embedding entire graphs in Euclidean space and computing meaningful distances between graphs, which can be used for both supervised and unsupervised learning, as well as exploratory analysis of graph-structured data.

Acknowledgements

References

Andén, J. and Mallat, S. Deep scattering spectrum. *IEEE Transactions on Signal Processing*, 62(16):4114–4128, August 2014.

Andén, J., Lostanlen, V., and Mallat, S. Classification with joint time-frequency scattering. arXiv:1807.08869, 2018.

Angles, T. and Mallat, S. Generative networks as inverse problems with scattering transforms. In *International Conference on Learning Representations*, 2018.

Atwood, J. and Towsley, D. Diffusion-convolutional neural networks. In *Advances in Neural Information Processing Systems 29*, pp. 1993–2001, 2016.

Borgwardt, K. M., Ong, C. S., Schönauer, S., Vishwanathan, S., Smola, A. J., and Kriegel, H.-P. Protein function prediction via graph kernels. *Bioinformatics*, 21(suppl_1):i47–i56, 2005a.

Borgwardt, K. M., Ong, C. S., Schönauer, S., Vishwanathan, S., Smola, A. J., and Kriegel, H.-P. Protein function prediction via graph kernels. *Bioinformatics*, 21(suppl_1):i47–i56, 2005b.

Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., and Vandergheynst, P. Geometric deep learning: Going beyond Euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

Brumwell, X., Sinz, P., Kim, K. J., Qi, Y., and Hirn, M. Steerable wavelet scattering for 3D atomic systems with application to Li-Si energy prediction. In *NeurIPS Workshop on Machine Learning for Molecules and Materials*, 2018. arXiv:1812.02320.

Bruna, J. and Mallat, S. Classification with scattering operators. In *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1561–1566, 2011.

Bruna, J. and Mallat, S. Audio texture synthesis with scattering moments. arXiv:1311.0407, 2013a.

Bruna, J. and Mallat, S. Invariant scattering convolution networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1872–1886, August 2013b.

Bruna, J. and Mallat, S. Multiscale sparse microcanonical models. arXiv:1801.02013, 2018.

Coifman, R. R. and Maggioni, M. Diffusion wavelets. *Applied and Computational Harmonic Analysis*, 21(1):53–94, 2006.

Cuesta, S. M., Rahman, S. A., Furnham, N., and Thornton, J. M. The classification and evolution of enzyme function. *Biophysical Journal*, 109(6):1082–1086, 2015.

Eickenberg, M., Exarchakis, G., Hirn, M., and Mallat, S. Solid harmonic wavelet scattering: Predicting quantum molecular energy from invariant descriptors of 3D electronic densities. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pp. 6540–6549, 2017.

Eickenberg, M., Exarchakis, G., Hirn, M., Mallat, S., and Thiry, L. Solid harmonic wavelet scattering for predictions of molecule properties. *Journal of Chemical Physics*, 148:241732, 2018.

Gama, F., Ribeiro, A., and Bruna, J. Diffusion scattering transforms on graphs. In *International Conference on Learning Representations*, 2019. arXiv:1806.08829.

Gatys, L., Ecker, A. S., and Bethge, M. Texture synthesis using convolutional neural networks. In *Advances in Neural Information Processing Systems 28*, pp. 262–270, 2015.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

Hammond, D. K., Vandergheynst, P., and Gribonval, R. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30:129–150, 2011.

Hirn, M., Mallat, S., and Poilvert, N. Wavelet scattering regression of quantum chemical energies. *Multiscale Modeling and Simulation*, 15(2):827–863, 2017. arXiv:1605.04654.

Kondor, R., Son, H. T., Pan, H., Anderson, B., and Trivedi, S. Covariant compositional networks for learning graphs. arXiv:1801.02144, 2018.

Kriege, N. M., Giscard, P.-L., and Wilson, R. On valid optimal assignment kernels and applications to graph classification. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29*, pp. 1623–1631. Curran Associates, Inc., 2016.

Lostanlen, V. and Mallat, S. Wavelet scattering on the pitch spiral. In *Proceedings of the 18th International Conference on Digital Audio Effects*, pp. 429–432, 2015.

Mallat, S. Group invariant scattering. *Communications on Pure and Applied Mathematics*, 65(10):1331–1398, October 2012.

Meyer, Y. *Wavelets and Operators*, volume 1. Cambridge University Press, 1993.

Niepert, M., Ahmed, M., and Kutzkov, K. Learning convolutional neural networks for graphs. In *International conference on machine learning*, pp. 2014–2023, 2016.

Oquab, M., Bottou, L., Laptev, I., and Sivic, J. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1717–1724, 2014.

Oyallon, E. and Mallat, S. Deep roto-translation scattering for object classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. arXiv:1412.8659.

Perlmutter, M., Wolf, G., and Hirn, M. Geometric scattering on manifolds. In *NeurIPS Workshop on Integration of Deep Learning Theories*, 2018. arXiv:1812.06968.

PyGSP. Graph signal processing in python (https://pygsp.readthedocs.io/en/stable/index.html), Accessed in September 2018.

Shervashidze, N., Vishwanathan, S., Petri, T., Mehlhorn, K., and Borgwardt, K. Efficient graphlet kernels for large graph comparison. In van Dyk, D. and Welling, M. (eds.), *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pp. 488–495, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, 2009. PMLR.

Shervashidze, N., Schweitzer, P., Leeuwen, E. J. v., Mehlhorn, K., and Borgwardt, K. M. Weisfeiler-Lehman graph kernels. *Journal of Machine Learning Research*, 12(Sep):2539–2561, 2011.

Sifre, L. and Mallat, S. Combined scattering for rotation invariant texture analysis. In *Proceedings of the ESANN 2012 conference*, 2012.

Sifre, L. and Mallat, S. Rigid-motion scattering for texture classification. arXiv:1403.1687, 2014.

Taheri, A., Gimpel, K., and Berger-Wolf, T. Learning graph representations with recurrent neural network autoencoders. In *KDD Deep Learning Day*, 2018.

Tixier, A. J.-P., Nikolentzos, G., Meladianos, P., and Vazirgiannis, M. Classifying graphs as images with convolutional neural networks. *arXiv preprint*, pp. arXiv:1708.02218, 2017.

Verma, S. and Zhang, Z.-L. Graph capsule convolutional neural networks. In *Joint ICML and IJCAI Workshop on Computational Biology*, 2018. arXiv:1805.08090.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

Yanardag, P. and Vishwanathan, S. Deep graph kernels. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1365–1374. ACM, 2015.

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems 27*, pp. 3320–3328, 2014.

Zhang, M., Cui, Z., Neumann, M., and Chen, Y. An end-to-end deep learning architecture for graph classification. In *AAAI Conference on Artificial Intelligence*, pp. 4438–4445, 2018.

Zou, D. and Lerman, G. Graph convolutional neural networks via scattering. arXiv:1804:00099, 2018.