Learnable Diffusion for Wavelets in Scattering Networks: Towards both Interpretability and Performance in Graph Representation Learning

Toan Van Tran[1] (✉) and Hung Son Nguyen[2]

[1] Hanoi University of Science and Technology, Hanoi, Vietnam.
toantranvan1203@gmail.com
[2] University of Warsaw, Warsaw, Poland. son@mimuw.edu.pl

Abstract. Scattering networks are deep convolutional architectures that use predefined wavelets for feature extraction and representation. They are mathematically well-understood, and have proven effective for classification tasks in limited training data scenarios, where traditional deep learning methods struggle. However, the opposite holds in larger data regimes, resulting in a performance gap between well-understood learning architectures and non-transparent yet highly effective paradigms. Our work addresses this gap on the domain of graphs by adapting the choice of diffusion operator that constructs the scattering network to the data, allowing better task-wise geometric representation. The resulting architecture preserves stability guarantees with respect to input perturbations. Continuous diffusion is applied in the learning process for more refined weight updates. Numerical experiments on benchmark datasets show that our approach consistently outperforms traditional graph scattering with predefined wavelets, expanding the scenarios where interpretable scattering architectures are competitive or superior to deep learning methods, and further reducing their aforementioned performance disparity.

Keywords: Graph learning · Scattering networks · Interpretability.

1 Introduction

Euclidean scattering networks are deep convolutional architectures analogous to Convolutional Neural Networks (CNNs). Unlike standard CNNs, which employ learnable filters at each layer, these networks are equipped with mathematically predefined wavelets selected from a multi-resolution filter bank ([15,3]). This distinction allows Euclidean scattering networks to serve as mathematically well-understood models that capture the principles underlying the empirical success of CNNs. Specifically, they exhibit proven robustness to small perturbations that are close to translations in the underlying domain ([3]). For classification, these models serve as efficient feature extractors, requiring only the classifier to be trained. This is especially beneficial with limited data, enabling state-of-the-art

performance while maintaining efficiency comparable to learned deep networks on simpler datasets.

The increasing focus on graph-structured data has spurred interest in adapting CNN architectures to these domains, leading to the development of effective graph convolutional models and variants (e.g. [14,26]). Naturally, proposal on extending the theoretical and practical benefits of Euclidean scattering networks to geometric data follows. [33] first introduced graph scattering networks using spectral wavelets ([13,22]) and analyzed its stability with respect to permutations of the nodes and perturbations on the spectrum of the underlying graph domain. Subsequently, [10] established improved stability bounds for this family of graph scattering transforms, applicable to more general graphs and independent of their spectral characteristics. Alternatively, [9] introduced graph scattering employing diffusion wavelets ([6]), using the lazy diffusion operator induced from normalized adjacency, and analyzing stability using diffusion metrics ([18,5]). Following this, [11] proposed an alternative graph scattering transform based on lazy random walk diffusion, demonstrating expressivity through extensive empirical evaluations.

A fundamental characteristic shared by all these scattering architectures is the use of fixed, often manually selected filters. This contributes to scattering networks' mathematical intepretability, and in low data scenarios helps them achieve higher classification performance than deep learning methods considered as black boxes. In larger data regimes, the performance of scattering architectures plateaus, while deep learning's becomes much higher than any predefined representations [20]. This work aims to bridge the performance gap between these interpretable and non-transparent learning paradigms in graph domains.

In particular, we consider diffusion graph scattering network [9], constructed from wavelets [6] which extract multiscales information in a single geometric diffusion process. Given a dataset, different diffusion operators can extract different properties via the use of diffusion map [7]. The selection of diffusion, which can be labor-intensive if manually done, is thus critical to the performance of the scattering network. Our approach for the diffusion scattering is thus to make the corresponding diffusion operator learnable, training it at the same time with the classifier. One operator is used throughout the network, making the number of additional parameters small. Our method also preserves expressivity and stability properties of the resulting architecture, maintaining the interpretability aspect of the original scattering.

The paper is organized as follows. In Section 2 we discuss related works. Section 3 provides the necessary background. Section 4 discusses the framework for defining diffusion wavelets and metrics (Sec. 4.1) and constructing the diffusion-based graph scattering transform (Sec. 4.2). Section 5 demonstrates the importance of diffusion operator selection with examples, introduces a learnable operator design (Sec. 5.1), establishes energy conservation bounds for wavelets (Sec. 5.2), provides a stability analysis of the resulting learnable diffusion scattering transform (Sec. 5.4), and complexity analysis (Sec. 5.5). Section 6 presents numerical results for graph classification tasks on low to medium data regimes.

2 Related Works

The performance gap between scattering architectures and deep learning methods in larger data regimes has been widely discussed, particularly in the context of Euclidean scattering, but remains less explored for graphs. For geometric data, [24] introduced a scale-adaptive extension of the lazy random walk diffusion scattering transform, enabling adaptive wavelet scale adjustment. Their approach demonstrated competitive performance compared to popular GNNs and the original graph scattering network.

For Euclidean image, [20] showed that the initial layers of a CNN can be replaced with a scattering network, forming a hybrid architecture that achieves competitive performance. [31] further demonstrated that learning in later CNN layers can be reduced to a dictionary matrix that computes a positive sparse l^1 code. Their model outperformed AlexNet on ImageNet 2012 while remaining mathematically interpretable. Additionally, [30] introduced a scattering-based model, in which only 1×1 convolutional tight frames are learned for scattering feature projection. This approach delivered performance comparable to ResNet-18. The authors of [12] investigated the role of non-linearity in deep CNNs and identified a phenomenon called "phase collapsing". They applied this into a Learned Scattering with 1×1 complex convolutional operators to achieve performance of ResNets of similar depths.

Adaptive diffusion for GNNs has also been explored in prior works [32,23]. These studies consider the weighted sum of outputs from each step of a predefined diffusion process as multiscale information, and propose learning these weights for adaptivity. In contrast, our approach focuses on adapting the diffusion operator, while multiscale feature extraction is handled by the scattering architecture using wavelets [6]. Unlike previous methods that emphasize multi-hop aggregation, we focus on improving how the diffusion process extracts information.

3 Preliminaries

We start with some background that will be used throughout the paper (most of which can be found in standard textbooks (e.g. [21,16])):

Metric space: A metric space is a tuple consisting of a set X and a distance function d, which satisfies the metric properties: $\forall x, y, z \in X$: (i) positivity: $d(x, y) > 0$, $\forall x \neq y$; (ii) reflexivity: $d(x, x) = 0$; (iii) symmetry: $d(x, y) = d(y, x)$, and (iv) triangle inequality: $d(x, y) \leq d(x, z) + d(z, y)$. A weighted undirected connected graph $G = (V, E, W)$, where W assigns positive weights to the edges, is an example of a metric space with the distance between two nodes x and y defined by $d(x, y) = \inf_{p_{x,y}} \sum_{e \in p_{x,y}} w_e$, where $p_{x,y}$ is a path connecting x and y.

Measure space: A measure space is a triple (X, Σ, μ), where X is a set, Σ is a σ-algebra on X (a nonempty collection of subsets of X closed under set-theoretic operations: complement, countable union, and countable intersection) and μ is a measure on (X, Σ). A **metric measure space** is a triple (X, d, m)

of a space X, metric d and a Borel measure m. For a finite graph G, mostly considered is the counting measure μ where $\mu\{u\} = 1,\ \forall u \in V$.

Multiresolution analysis: A multiresolution analysis of $\mathcal{L}^2$ of a metric measure space (X, d, μ) is a sequence of subspaces $\{V_j\}_{j\in\mathbb{Z}}$, each of which is called an **approximation space**. In the case of $\mathcal{L}^2(\mathbb{R})$, the sequence $\{V_j\}_{j\in\mathbb{Z}}$ satisfies the properties:

(i) $\lim_{j\to-\infty} V_j = \bigcup_{j=-\infty}^{+\infty} V_j = \mathcal{L}^2(\mathbb{R})$
(ii) $\lim_{j\to+\infty} V_j = \bigcap_{j=-\infty}^{+\infty} V_j = \{0\}$
(iii) $V_{j+1} \subseteq V_j,\ \forall j \in \mathbb{Z}$
(iv) There exists a Riesz basis that spans V_0.

The **detail space** W_j is defined as the orthogonal complement of V_j in V_{j-1}; in other words, $V_{j-1} = V_j \oplus^{\perp} W_j,\ \forall j \in \mathbb{Z}$. The orthogonal projection of a signal x on V_{j-1} can thus be decomposed as $P_{V_{j-1}}x = P_{V_j}x + P_{W_j}x$. The projection of a signal x on W_j captures the "details" of x that are present in the finer-scale space V_{j-1} but absent in the coarser-scale V_j. Given a mother wavelet ψ, the translations of ψ after being dilated onto scale 2^j, denoted as $\{\psi_{j,n}\}_{n\in\mathbb{Z}} = \{\frac{1}{\sqrt{2^j}}\psi(\frac{t-2^j n}{2^j})\}_{n\in\mathbb{Z}}$, compose an orthonormal basis of W_j. On said basis, the projection of x on W_j can be obtained by a partial expansion: $P_{W_j}x = \sum_{n=-\infty}^{+\infty}\langle x, \psi_{j,n}\rangle\psi_{j,n}$.

Scattering transform is a mapping which takes an input signal x and returns a representation $\Phi(x)$, calculated based on a deep convolutional architecture, stable to small deformations while preserves high-frequency information. $\Phi(x)$ is computed by applying sequentially three elements: A **filter bank** of band-pass wavelets $\{\{\psi_{j,k}\}_{k=0}^{K-1}\}_{j=1}^{J}$, a **pointwise nonlinearity** ρ (modulus or ReLU), and an **average operator** U. In the Euclidean setting, the filter bank consists of rotated and dilated versions $\psi_{j,k}$ of a mother wavelet ψ with scaling parameter j and angle parameter k, with the angle $\theta \in \{2\pi k/K\}_{k=0,\ldots,K-1}$. The scattering representation of x is defined as:

$$\begin{aligned} \Phi(x) &= \left[S_0(x), S_1(x), \ldots, S_{m-1}(x)\right], \text{ where} \\ S_k(x) &= \left[U\Pi_{i=0}^{k}\left(\rho\psi_{\alpha_i}\right)(x)\right]_{\alpha_0,\alpha_1,\ldots,\alpha_k} \\ &= \left[U\left(\rho\left(\ldots\rho\left(\rho\left(x * \psi_{\alpha_0}\right) * \psi_{\alpha_1}\right)\ldots * \psi_{\alpha_k}\right)\right)\right]_{\alpha_0,\alpha_1,\ldots,\alpha_k}. \end{aligned} \tag{1}$$

where $\alpha_i,\ i = 0, ..., k$ represent the scale parameters.

4 Graph Diffusion Scattering Transform

4.1 Graph Diffusion Wavelets and Diffusion Distances

The works in [8,6] introduce a framework for multiscale and multiresolution analysis on the domain of graphs, based on polyadic powers of a diffusion operator. We consider an undirected, weighted, and connected graph $G = (V, E, W)$, with $|V| = n$ nodes, edges set E and adjacency matrix $W \in \mathbb{R}^{+^{n\times n}}$. The random

walk matrix $T = WD^{-1}$ of G defines an induced diffusion process on its nodes, where $D = \text{diag}(d_1, ..., d_n)$, and $d_i, i = 1, ..., n$ are the degrees of the nodes of G. For stability, the lazy diffusion $P = \frac{1}{2}(I + T)$ can be employed. Given that P is left-stochastic and guaranteed to have postive entries at indices (u, v) whenever $(u, v) \in E$, it can also be interpreted as a transition matrix of a random walk process on G.

The operator P is mass-preserving (i.e. $\sum_{(u,v)\in E} P[v, u] = 1$ for any fixed u), contractive ($\|P\| \leq 1$), and positivity-preserving ($x \geq 0 \Rightarrow Px \geq 0$). Consider a random walk on G with P as the transition matrix, the probability distribution starting from an initial p_0 (e.g. a Dirac delta δ_u at any node u of G) becomes increasingly "smoothed out" as over time, as observed from the fact that $P^t p_0$ converges to a stationary distribution when $t \to \infty$, and this distribution is independent of p_0.

Based on this "smoothening" property, P can be interpreted as a dilation operator, acting on signals on $\mathcal{L}^2(G)$. An analog to the multiresolution analysis can thus be constructed, as proposed in [6]. In a more general perspective, we consider a diffusion semigroup $\{A^t\}_{t\geq 0}$ induced by a general diffusion operator A acting on $\mathcal{L}^2(X, \mu)$ which satisfies the following properties:

(i) $\|A^t\|_p \leq 1$, for every $1 \leq p \leq +\infty$.
(ii) $A^t x \geq 0$, for every $x \geq 0$

Semigroups as such are referred to as Markovian semigroups. We fix a precision level $\epsilon < 1$. Define $A\mathcal{L}^2(X) = \text{span}\{x \in \mathcal{L}^2(X) : \|x\| \leq 1, \frac{\|Ax\|}{\|x\|} \geq \epsilon\}$. Let $\lambda_{min} = \inf_{x\in\mathcal{L}^2(X),\ \|x\|\leq 1} \frac{\|Ax\|}{\|x\|}$, and $\lambda_{max} = \sup_{x\in\mathcal{L}^2(X),\ \|x\|\leq 1} \frac{\|Ax\|}{\|x\|}$. As $\|A\| \leq 1$, it follows that $\dim(A\mathcal{L}^2(X)) \leq \dim(\mathcal{L}^2(X))$. The operator A contracts the functional space $\mathcal{L}^2(X)$ after each application. The inequality may be strict, as there are signals in some parts of $\mathcal{L}^2(X)$ have their norm contracted by λ_{min}, which may already be smaller than ϵ.

At times $t_j = \gamma^{j+1}$, where $\gamma > 1$ (commonly set to 2), we discretize $\{A^j\}$ following classical wavelet theory, having wavelets are dilated at scales of polyadic powers. We define the approximation spaces V_j analogous to a multiresolution analysis of $\mathcal{L}^2(X)$ as $A^{t_j}\mathcal{L}^2(X)$. We also conventionally define $V_{-1} = \mathcal{L}^2(X)$. A family of multiresolution filters, analogous to the wavelets filter bank in the Euclidean setting, can thus be defined as:

$$\psi_0 = I - A \ , \ \psi_i = A^{t_{i-2}} - A^{t_{i-1}} = A^{2^{i-1}} - A^{2^i} (i > 0) \tag{2}$$

These filter can be understood as projecting a signal x onto the complement of V_j in V_{j+1}, analogous to the partial expansion of x in the wavelet basis $\{\psi_{j,n}\}$ of W_j ([9]), thereby extracting the details of x at coarser scales as j increases.

A diffusion distance can also be constructed on the operator A ([5]). If A is left-stochastic (i.e. it can be considered as a transition matrix of a Markov chain) and positivity-preserving, then the diffusion distances at time t between two nodes u and v is given by: $d_t(u, v) = \|A^t\delta_u - A^t\delta_v\|$, with the norm induced from the inner product weighted by $1/\pi_A$. This distance considers all path of

length t between u and v. If there are many connecting short paths between the two nodes, then $d_t(u,v)$ will be small. It is, as a consequence, robust to noise, unlike the shortest path distance. An additional consequence is that $d_t(u,v)$ is small if u's and v's neighborhoods are similar.

A distance between two graphs of equal size can also be defined based on this node-level one. Given two graphs $G = (V, E, W)$ and $G' = (V', E', W')$ with $|V| = |V'| = n$ and respective symmetric diffusion operators A_G and $A_{G'}$, the normalized diffusion distance between G and G' at time t is defined in [9] as:

$$\tilde{d}_t(G, G') = \inf_{\Pi \in \Pi_n} \|(A_G^t)(A_G^t)^* - \Pi^{-1}(A_{G'}^t)(A_{G'}^t)^*\Pi\| \tag{3}$$

where Π_n is the space of all $n \times n$ permutation matrices, A^* is the adjoint of operator A. $(A_G^t)(A_G^t)^*$ is the Gram matrix of the system $A_G^t \delta_{u\, u \in V}$. The distance thus compares the 2 vector systems intrinsic to G and G' at time t, and is invariant to permutation and orthonormal transformation. It is also robust to noise similarly to the node-level one. For simplicity, we consider $t = 1$ in this work. As random walk matrices are not generally symmetric in the same inner product, a weighted variant of (3) can be considered:

$$\begin{aligned} d(G, G') &= \inf_{\Pi \in \Pi_n} \|(A_G) D_{A_G} (A_G)^* - \Pi^{-1}(A_{G'}) D_{A_{G'}} (A_{G'})^* \Pi\| \\ &= \inf_{\Pi \in \Pi_n} d(G, G', \Pi) \end{aligned} \tag{4}$$

where $D_A = \mathrm{diag}(\pi)$ where π is the limiting distribution of A.

Each entry of $(A_G^2)(A_G^2)^*$ is $(W^2)_{u,v}/\deg(u)\deg(v)$, representing a form of local normalization, thus using (3) focuses on structural equivalence. In contrast, a global normalization takes the form $(W^2)_{u,v}/\mathrm{vol}(G)^2$, where $\mathrm{vol}(G) = \sum_{u \in V} \deg(u)$, giving more weights to important nodes (i.e. those with high degree). The distance in (4) normalizes by $\sqrt{\deg(u)\deg(v)}\mathrm{vol}(G)$, thereby balancing between the two. In this work, we consider the distance on graphs of equal sizes; however, it can be naturally extended to graphs of different sizes by replacing permutation matrices with soft-correspondences, as in [2] ([9]).

4.2 Graph Scattering Transform

The construction of the multiresolution analysis, and thus an analog of the wavelets filter bank on the domain of graphs, paves the way for the extension of graph scattering transform. Let $\Psi_n : \mathcal{L}^2(X) \to (\mathcal{L}^2(X))^{J_n}$ be the wavelet decomposition operator that maps x to $(\psi_j x)_{j=0,\ldots,J_n-1}$, with ψ_j defined as in the previous subsection. Following the Euclidean setting described in Section 3, the diffusion graph scattering transform $\Phi_G(x)$ is also defined from three components: the wavelet decomposition operator at each layer k: Ψ_k; a pointwise nonlinearity ρ; and a low-pass operator U. The representation $\Phi(x)$ is calculated analogously to the scattering transform in Section 1 (see Figure 1).

In [9], $\Phi_G(x)$ is introduced with the multiresolution filters being constructed from the intrinsic lazy normalized symmetric adjacency $\overline{P} = \frac{1}{2}(I + M)$ of $G =$

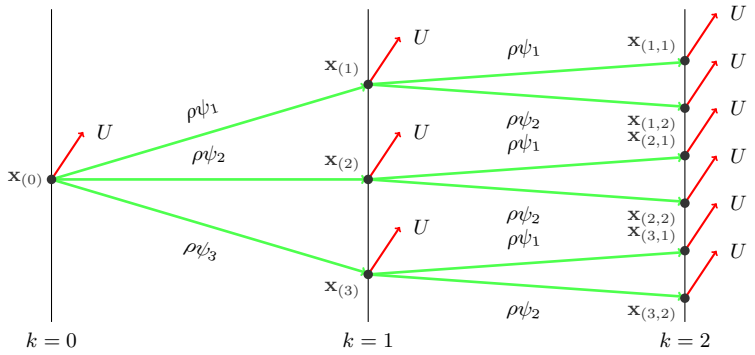

Fig. 1: Illustration of graph scattering transform with $m = 3$ layers, scales $J_1 = 3$ and $J_2 = 2$.

(V, E, W), where $M = D^{-1/2}WD^{-1/2}$. Although $\overline{P}$ is not mass-preserving, there is a spectral theory to this operator, with respect to the canonical inner product on $\mathcal{L}^2(G)$. This is desirable in many cases - for example, when constructing a diffusion embedding such that the Euclidean distance in the embedding space corresponds to the diffusion distance in the original graph space [7]. Moreover, since $\overline{P}$ is contractive (due to its self-adjointness and having spectral radius $\rho(\overline{P}) \leq 1$) and positivity preserving, the multiresolution analysis construction remains valid. The average operator U is taken to be the infinite-time diffusion limit $\lim_{t\to\infty} \overline{P}^t$, expressible as $Ux = \langle \mathbf{v}^\mathsf{T}, x\rangle$, where $\mathbf{v} = \frac{\mathbf{d}^{1/2}}{\|\mathbf{d}^{1/2}\|_2} = (\frac{\mathbf{d}}{\|\mathbf{d}\|_1})^{1/2}$ is the eigenvector of $\overline{P}$ corresponding to the eigenvalue 1, $\mathbf{d}$ being the degree vector of G, and $\mathbf{x}^{1/2}$ is the vector with square root of every entry of $\mathbf{x}$.

5 Graph Scattering Networks with Learnable Diffusion

In this section, we first discuss examples of how different diffusion kernels can capture different properties of a dataset via the use of diffusion map, as part of our motivation. We then propose a formulation of a learnable diffusion operator that we will use in our experiments as an example to demonstrate the enhanced performance. In subsequent parts, expressivity and stability bounds for wavelets and scattering networks constructed from said operator are established.

5.1 Learnable Diffusion Kernels and Diffusion Operators

We consider some examples of X approximately lying on some submanifolds $\mathcal{M}$ of $\mathbb{R}^n$, characterized by a density function $p(x)$, to show how different kernels can captures different properties, e.g. the intrinsic geometry of the data points, its distribution density, or a combination of both ([7]). This feature extraction is done via the use of diffusion map [5]. Between two points x, y of X, let $k(x, y)$ be

the "affinity function" that is symmetric, positivity-preserving, and positive semi-definite. $k(x, y)$ can be interpreted as the analog of edges weight between graph nodes. Let $d(x) = \int_X k(x, y)\mu(y)$ be an analog of degrees of nodes, where μ is a probability measure. The random walk diffusion operator A can thus be defined as $As(x) = \int_X a(x, y)s(y)d\mu(y)$, where s is a signal on X, and $a(x, y) = \frac{k(x,y)}{d(y)}$.

Two examples of diffusion kernels are given in [7], one accounting for the density of the points in X, and the other captures the geometry irrespective of density. Consider the random walk diffusion A_ϵ constructed from an isotropic kernel $k_\epsilon(x, y) = \exp(-\|x - y\|^2/\epsilon)$. If $p(x)$ is uniform, A_ϵ approximate the Laplacian-Beltrami operator Δ on $\mathcal{M}$, as $\epsilon \to 0$ ([1]). On the other hand, if $p(x)$ is not, A_ϵ tends to a more general operator of the form $\Delta+Q$, where $Q(x) = \frac{\Delta p(x)}{p(x)}$ acts as a potential term, reflecting the influence of the non-uniform density.

An alternative normalization is introduced that captures the geometry of the data points by taking into account the non-uniformity of $p(x)$: Let $p_\epsilon(x) = \int_X k_\epsilon(x, y)p(y)dy$, and define the new kernel $\hat{k}_\epsilon(x, y) = k_\epsilon(x, y)/p_\epsilon(x)p_\epsilon(y)$. The corresponding random walk diffusion $\hat{A}_\epsilon$ then serves as an approximation of the Laplace-Beltrami operator at time ϵ, regardless of density variations.

These examples show that the embeddings obtained are highly sensitive to the choice of kernel. Depending on the task, one may prefer this diffusion to another. For example, the second kernel discussed above are used for segmentation with spectral clustering [28], while the first one can be used for analysis solely on the topology of the domain. Thus, there are cases where data-driven diffusion is naturally preferable.

For each node u of a graph G, we define the **descriptor** g_u to be a vector that has the characteristics of u, e.g. its node degree. Between every two adjacent nodes u and v, let $k(u, v)$ be the kernel that quantifies the affinity between the two, being positive if u and v are adjacent. Taking inspiration from attentional diffusion in [4], we propose the affinity kernel between two distinct, adjacent nodes to be given by:

$$k(u, v) = \exp\left(\frac{\langle W(g_u), W(g_w)\rangle}{\|W(g_u)\|.\|W(g_v)\|}k_1\right) \tag{5}$$

where $\|\cdot\|$ is the vector norm, W is a mapping from the descriptor space $\mathcal{G}$ to an embedding space $W(\mathcal{G})$, and k_1 is a hyperparameter to be tuned. This formulation differs from the affinity used in scale-dot attention ([25]), which is given by $k_{sd}(u, v) = \exp(\frac{(W_K g_u)^\intercal W_Q g_v}{d_e})$, in two key aspects: First, k is symmetrized by letting $W_K = W_Q$, where both mappings can be nonlinear transformations (e.g. a simple MLP), thereby preserving generalization capability. Second, the inner product is normalized to be the cosine-similarity. In our experiments, we found out that the resulting attention weights without normalization tend to be "extreme", i.e. one neighbor would dominate, causing the attention values to be reduced to either 0 or 1. As cosine similarity is at most 1, we introduce a relaxing hyperparameter $k_1 \in [0, \infty)$, to extend the possible magnitude range

of the affinity function. One can apply random walk normalization to k to construct the diffusion kernel. However, for stability and convergence reasons, we reformulate the diffusion kernel $a(u, v)$ between any two nodes (either adjacent or identical) as follows:

$$\begin{aligned} a(u,v) &= \big[k(u,v)/K_u\big] * \big[\sigma(\alpha(u)) * (1-k_2) + k_2/2\big] \text{ if } u \neq v, \\ a(u,u) &= 1 - \big[\sigma(\alpha(u)) * (1-k_2) + k_2/2\big] \end{aligned} \tag{6}$$

where $K_u = \sum_{v \in \mathcal{N}(u)} k(u, v)$, σ denotes the sigmoid function, $k_2 \in [0, 1]$ is an additional hyperparameter, and $\alpha(u) = \langle W(g_u), \alpha \rangle$, with α is a learnable vector of dimension $\dim(W(\mathcal{G}))$. We introduce α to also allow learnability into self-diffusion, which is necessary for the convergence of the diffusion process. k_2 here is used to control the possible range of $a(u, u)$, thereby preventing it becomes too "extreme". We would like to remark that k_1 and k_2 can be interpreted as regularization hyperparameters, as setting $k_1 = 0$ and $k_2 = 1$ recovers the standard random walk diffusion kernel for the unweighted version of the graph G.

Let $\mathbf{A} : \mathcal{G} \to (\mathcal{L}^2(G))^2$ be the operator which maps g to a diffusion matrix A of g. By definition, A is left-stochastic. To enhance stability during the training process, we employ a multi-head attention mechanism analogous to that introduced in ([25,26]) by taking the average across the heads: $\mathbf{A}(g) = \frac{\sum_{k=0}^{h-1} \mathbf{A}_k(g)}{h}$. This matrix can then be used directly as diffusion operator $A = \mathbf{A}(g)$.

However, modeling the diffusion in graph neural networks (GNNs) as a continuous-time process has been shown to enhance both training stability and performance ([27]). The same approach could thus be done for the above formula. One could discretize update step between two consecutive powers of A by taking fractional temporal difference. Temporal discretization schemes for continuous process, such as Euler or Runge-Kutta, can be used for such purpose. A quick discussion of these schemes is given in the supplementary materials [1]. Further experiments are presented and discussed in Section 6.

5.2 Wavelets with Learnable Diffusion

The construction of wavelets, in general, relies on the framework of multiresolution analysis, which we have mentioned in Section 4.1. For such construction to be possible, the diffusion operator used must have a single limiting distribution. This condition is satisfied, as our diffusion operator A above is irreducible (since the underlying domain G is connected and $a(u, v) > 0$ if $(u, v) \in E$) and aperiodic ($\exists u : A(u, u) > 0$). This is a basic result from the theory of Markov processes.

Having our multiresolution analysis, constructed using A with the filters in Section 4.1, we can now obtain a wavelet decomposition operator Ψ for the proposed adaptive scattering network. We now prove Ψ is a frame analysis operator, i.e. it defines a frame. This ensure expressivity guarantees on the representation

[1] Supplementary materials are provided at https://github.com/toanvtran/learnable-diffusion-scattering

returned by Ψ. Notationally, $\langle x, y\rangle_{D_A} = x^T D_A y$ is a weighted inner product, where $D_A = \text{diag}(\pi_A)$ and π_A is the stationary distribution of A. $\|\cdot\|_{D_A}$ refers to the weighted ℓ^2-norm induced by this inner product. $\langle\cdot\rangle$ and $\|\cdot\|$ refers to the canonical inner product and ℓ^2-norm, respectively.

Proposition 1. *On a connected domain G, let Ψ be the wavelet decomposition operator on $\mathcal{L}^2(G)$ based on the non-negative matrix A defined as above. Assume that for every $x \in \mathcal{L}^2(G)$ satisfying $\langle x, \pi_A\rangle_{D_A} = 0$, $\frac{\|Ax\|_{D_A}}{\|x\|_{D_A}} < 1$. Let $\beta_A = \inf_x(1 - \frac{\|Ax\|_{D_A}}{\|x\|_{D_A}})$. Then, there exists constants $M(\beta_A)$, $N(\beta_A) > 0$ depending only on β_A such that for any x as above:*

$$M(\beta_A)\|x\|^2_{D_A} \leq \sum_{j=0}^{J-1} \|\psi_j x\|^2_{D_A} \leq N(\beta_A)\|x\|^2_{D_A} \tag{7}$$

The proof is presented in the supplementary materials. The existence of the two bounds is a necessary and sufficient condition that there exists a bounded inverse for each decomposition on the image space $\text{Im}(\Psi)$. This means Ψ defines on $\mathcal{L}^2(G, \mu_{\pi_A})$ a complete and stable representation.

According to the general Perron-Frobenius theory, any irreducible and aperiodic matrix A with non-negative elements has a unique eigenvector π_A corresponding to its largest eigenvalue, 1, up to a constant multiple. Furthermore, the remaining eigenvalues of A, considered in the unitary space, have strictly smaller moduli. However, there is no guarantee that the orthogonal complements M_{π_A} of $\text{span}(\pi_A)$ in $\mathcal{L}^2(G)$ will remain invariant under the action of A. As every signal which is a multiple of π_A lose all of its information under the wavelet decomposition, to prevent unnecessary information loss, we would want to design A such that $AM_{\pi_A} \subseteq M_{\pi_A}$. A straightforward family of matrices satisfying this property is the class of self-adjoint matrices. Ensuring symmetry in the affinity function k, as in our construction, is a sufficient condition for this.

It is also worth noting that the condition that G be connected can be relaxed. Specifically, G can consist of p connected components that are pairwise disconnected, provided $p \ll |V| = n$. This condition is necessary because each component can have its own stationary distribution, making the subspace of stationary distributions of A on $\mathcal{L}^2(G)$ of multiple dimensions, with a maximum dimension of p. Any signal in this subspace will lose all of its information upon applying Ψ, thus rendering Ψ useless for such signals. For simplicity, we continue to consider the case where G has only 1 connected component.

5.3 Graph Scattering Transform with Learnable Diffusion

We construct our adaptive variant of Graph Scattering Transform similarly to the one in Section 4.2 by replacing the fixed decomposition operator with the adaptive version we defined above. Additionally, we employ the average mean pooling operator U, which is independent of A: $Ux = \langle \mathbf{1}/\mathbf{n}, x\rangle$. In particular, on

a connected graph G with a graph signal x, the transformation at each layer is given by:

$$\begin{aligned}\phi_k = U(\rho\Psi)^k x &= \left[U\Pi_{i=0}^{k}\left(\rho\psi_{j_i}\right)(x)\right]_{j_0,j_1,\ldots,j_k} \\ &= \left[U\rho\psi_{j_k}\ldots\rho\psi_{j_1}\rho\psi_{j_0}x\right]_{j_0,j_1,\ldots,j_k}.\end{aligned} \tag{8}$$

where $\{\psi_{j_i}\}_{j_i}$ are multiresolution filters constructed using the adaptive operator A. Thus, the scattering representation obtained from an m-layer network is:

$$\Phi(x) = [Ux, \phi_1(x), \ldots, \phi_{m-1}(x)] = \left[Ux, U\rho\Psi x, \ldots, U(\rho\Psi)^{m-1}x\right] \tag{9}$$

In the following we provide the stability analysis of the adaptive graph scattering transforms using our learnable diffusion operators:

5.4 Stability Analysis

A robust and meaningful signal representation should exhibit stability to noise, meaning that a small change in the input signal yields proportionally small variations in the output representation. As mentioned in Section 4.1, the matrix A, being a random walk operator, naturally induces a graph-level diffusion distance. We begin by establishing the stability of the wavelet decomposition operator in the following lemma.

Lemma 1. *On two distinct graphs G and G' with $|V| = |V'| = n$, let Ψ_G and $\Psi_{G'}$ be the wavelet decomposition operators induced from respectively A_G and $A_{G'}$. Consider all signal x with both $\frac{\|A_G x\|}{\|x\|}$ and $\frac{\|A_{G'}x\|}{\|x\|} < 1$. Let $\beta = \min\{\inf_x(1 - \frac{\|A_G x\|}{\|x\|}), \inf_x(1 - \frac{\|A_{G'}x\|}{\|x\|})\}$. Let $\delta_\pi = \max\{\min_i\{\pi_{A_G,i}\}, \min_j\{\pi_{A_{G'},j}\}\}$. Assume that the spectra of A_G and $-A_{G'}$ are disjoint, where every pair of eigenvalues are at least δ from each other. We have:*

$$\inf_{\Pi\in\Pi_n} \|\Psi_G - \Pi\Psi_{G'}\Pi^{\mathsf{T}}\| \leq C_{A_G,A_{G'}} d_1(G, G') \tag{10}$$

where $d_1(G, G') = \inf_{\Pi\in\Pi_n}\left[d(G, G', \Pi) + (1-\beta)^2\|\pi_{A_G} - \Pi\pi_{A_{G'}}\Pi^{-1}\|_\infty\right]$, $C_{A_G,A_{G'}} = \frac{\sqrt{2C_1+4C_2}}{\delta_\pi}$, $C_1 = n\frac{\kappa(D_{A_G})\kappa(D_{A_{G'}})}{\delta}^2$, $C_2 = \frac{(1-\beta)^2(2-2\beta+\beta^2)}{(2\beta-\beta^2)^3}$, $D_A = \mathrm{diag}(\pi_A)$, $\kappa(D_A) = \sqrt{\frac{\max_i \pi_A}{\min_i \pi_A}}$, and $d(G, G', \Pi)$ as presented in Sec. 4.1.

The complete proof is presented in the supplementary materials. Since A_G and A'_G may not be symmetric with respect to the same weighted inner product, an additional term is introduced to measure the discrepancy between their stationary distributions. If this discrepancy is small, then the bound can be characterized as linear in $d(G, G')$, which is discussed in Sec. 4.1. This lemma serves as the primary tool in proving the next result, which establishes stability bounds for an m-layer graph scattering network under small perturbations to the graph structure:

Theorem 1. *Let $x \in \mathbb{R}^n$ and $\Phi_G(x)$ be the m-layer scattering representation of a signal x on a graph G, and let $\Phi_{G'}(x)$ be the same respectively on graph G'. With the same assumption and notation as in Lemma 1, let $N = \max\{N(\beta_{A_G})\kappa(D_{A_G}), N(\beta_{A_{G'}})\kappa(D_{A_{G'}})\}$, $N(\beta_A)$ be as in Proposition 1. We have:*

$$\|\Phi_G(x) - \Phi_{G'}(x)\|^2 \leq \sum_{k=0}^{m-1} \left[kN^{k-1}C_{A_G,A_{G'}}d_1(G,G')\right]^2 \|x\|^2 \tag{11}$$

The proof is presented in detail in the supplementary materials. Theorem 1 provides the stability bound for the scattering representations of the same signal x on two different graphs G and G'. Each graph has its own multiresolution analysis, and if the distance $d_1(G, G')$ between the two graphs is small, then the discrepancy between the resulting representations will also be small. Since $m \leq 5$ in most applications (as the scattering energy rapidly diminishes in deeper layers with increasing m ([3])), the change in the learnable scattering representations due to a small topological perturbation is effectively characterized by a linear dependence on $d_1(G, G')$.

5.5 Complexity

Number of parameters: In this work, we adopt the traditional architecture of the scattering transform, where the same wavelet decomposition operator is used throughout, and all of its wavelets are generated from a single mother wavelet. Consequently, only one filter needs to be learned across the entire scattering network. The additional number of parameters compared to traditional scattering is $\mathcal{O}(KPH)$, where K is the size of each descriptor, P is the number of parameters in the mapping W, and H is the number of heads. This does not depend on the size of the scattering network or the size of the graph G.
Memory requirement: We consider a scattering network of m layers, and each layer has k wavelets. Since the model has to store the attributes in each wavelet scale for doing low-pass averaging and diffusion in subsequent layer, the memory requirement is $\mathcal{O}(Ck^mN)$, where C is the number of input channels, $N = |V|$ is the number of graph nodes. Since m and k are predefined hyperparameters, with $m \leq 5$ in most applications, the memory requirement effectively scales linearly with the number of graph nodes.

6 Numerical Experiments

In this section, we empirically demonstrate the discriminative power of the scattering transform with learnable diffusion in classification tasks on two types of datasets: social networks and bioinformatics, particularly in low- to medium-data regimes. Our results show the method extends the scenarios where interpretable models are competitive to non-transparent deep learning methods in term of performance.

Table 1: Classification accuracies as a function of percentage of training data used in the social network dataset (IMDB-BINARY) and the bioinformatics dataset (MUTAG). The highest, second-highest, and third-highest accuracies are highlighted in blue, orange, and red, respectively.

		Deep learning		**Traditional scattering**		**Ours**
	Training amount*	**GIN-0 (MLP-sum)**	**UGformer**	**GS-SVM**	**GSN +MLP**	**LD-GSN +MLP**
IMDB-BINARY	**1%$_{(10)}$**	58.52 ± 5.37	56.96 ± 2.33	**59.81 ± 5.27**	**60.21 ± 4.17**	**63.03 ± 3.70**
	2.5%$_{(25)}$	**63.45 ± 7.18**	60.71 ± 3.63	61.27 ± 3.45	**62.79 ± 3.15**	**65.17 ± 3.20**
	5%$_{(50)}$	**65.40 ± 2.07**	64.52 ± 2.15	61.88 ± 1.98	**64.70 ± 2.40**	**66.83 ± 1.96**
	7.5%$_{(75)}$	**67.63 ± 1.47**	**65.99 ± 1.93**	63.45 ± 1.68	64.84 ± 1.64	**68.15 ± 2.06**
	10%$_{(100)}$	**68.36 ± 1.52**	**67.92 ± 0.94**	65.62 ± 2.93	65.15 ± 1.99	**68.58 ± 1.02**
	20%$_{(200)}$	**70.51 ± 0.97**	**70.20 ± 0.92**	66.46 ± 1.56	66.56 ± 3.43	**70.90 ± 4.63**
MUTAG	**2%$_{(3)}$**	**70.90 ± 3.24**	68.82 ± 7.35	70.08 ± 2.74	**70.55 ± 2.72**	**71.85 ± 4.36**
	2.5%$_{(4)}$	**71.78 ± 3.40**	69.28 ± 8.93	**71.65 ± 2.95**	71.11 ± 1.98	**72.87 ± 2.52**
	5%$_{(9)}$	**75.51 ± 3.20**	72.41 ± 2.25	72.84 ± 3.56	**74.86 ± 3.47**	**77.08 ± 3.85**
	7.5%$_{(14)}$	**77.89 ± 3.07**	**76.84 ± 3.15**	75.24 ± 2.61	75.63 ± 3.17	**79.41 ± 2.77**
	10%$_{(18)}$	**79.04 ± 3.81**	**78.91 ± 2.78**	75.47 ± 2.61	77.24 ± 3.80	**80.83 ± 3.17**
	20%$_{(37)}$	**82.98 ± 2.34**	**80.50 ± 3.13**	77.11 ± 2.27	79.51 ± 2.22	**81.51 ± 2.36**

* Percentage of dataset used for training, followed by the actual number of samples

We conduct experiments on well-known social network and bioinformatics datasets as described in [17]. To maintain consistency with our theoretical framework, we restrict our experiments to 2 datasets comprising of connected graphs, namely, IMDB-BINARY and MUTAG, and perform graph-level classification on them with varying amount of training data. A detailed description of these datasets is provided in the supplementary materials. For each node u, the descriptor g_u is chosen to be a vector consists of topological features of u and its neighborhood: degree, eccentricity, clustering coefficient, number of triangles contains u as a vertex, core number, clique number, and PageRank. While MUTAG provides intrinsic node features, which can be used as input to the diffusion process, we use the descriptors as proxies for the featureless IMDB-BINARY dataset. Some discussion on the alternative usage of node features in place of descriptor can also be found in the supplementary materials.

For the classification task, we employ a model that integrates our learnable diffusion graph scattering network as a feature extractor with a simple MLP as the classifier, denoted as LD-GSN+MLP. Using the MLP allows the learning of the kernel weights in our model via backpropagation. As a baseline, we implement the same architecture but with a lazy random walk operator $\frac{1}{2}(I+WD^{-1})$, referred to as GSN+MLP. We compared our method against traditional scatter-

ing methods (GSN+MLP, GS-SVM [11]), graph transformer (UGformer [19]), and graph neural network (GIN-0 (MLP-sum) [29]). These models were chosen for their publicly available implementations.

Performance: Table 1 presents classification accuracy as a function of training data percentage, with the rest used for validation. Further experimental details are provided in the supplementary materials. As expected, increasing training data generally improves accuracy across all models. LD-GSN+MLP consistently outperforms competitors, whether when traditional scattering surpasses deep learning or when deep models regain dominance with more data. The only exception is MUTAG at 20% training data, where deep learning begins to recover its advantage. LD-GSN ranks second, with a statistically minimal gap to the top-performing GIN-0.

LD-GSN benefits from the stability of scattering networks while improving adaptability through learnable diffusion, regulated via parameters introduced in Sec. 5.1. This adaptivity, shown by our experiments, results in a consistent performance advantage over both deep learning and traditional scattering approaches in low to medium data regimes, further expanding the applicability of interpretable models for graph data.

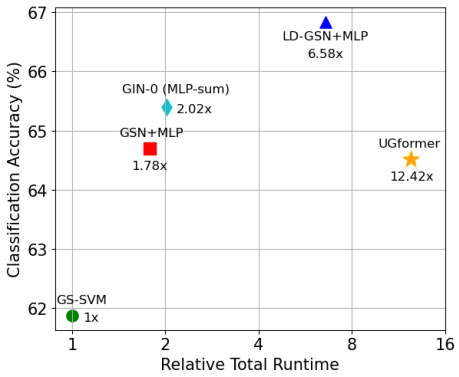

Fig. 2: Total running time versus classification accuracy on IMDB-BINARY with 2.5% data for training.

Running time: We compare the end-to-end runtime of the five models using 2.5% of IMDB-BINARY as training data on an NVIDIA A100 40GB GPU (Figure 2, logarithmic scale). GIN-0, UGformer, and LD-GSN+MLP require adding node features, while GS-SVM and GSN+MLP also extract scattering representations. Neural networks train for 200 epochs, whereas GS-SVM is fitted once. LD-GSN+MLP runs $\approx 3.5\times$ slower than GSN+MLP and GIN-0 due to backpropagation through the scattering architecture but achieves significantly higher accuracy.

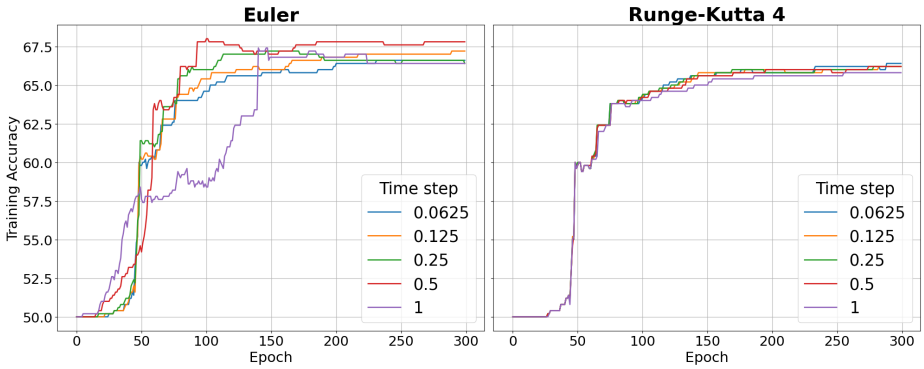

Fig. 3: Training accuracies as a function of epoch for different time step and approximation scheme choices on IMDB-BINARY with otherwise same configuration and initialization.

Effect of temporal discretization schemes: As mentioned in Sec. 5.1, the diffusion process can be modeled as a continuous one. We perform additional experiments to investigate how the choice of time step or temporal discretization schemes affect the stability of the training process (Figure 3). Increasing the time step or employing discretization schemes with higher numerical accuracy improves the numerical precision of each weight update, resulting in a more stable and refined training curve, similarly to adjusting the learning rate. However, in our case, due to the highly non-convex nature of the optimization problem, this does not necessarily translate to better performance as observed in [27] for linear GCNs. A balance should be achieved between stability and the ability to escape local minima. Consequently, we treat the time step as a hyperparameter in our experiments.

7 Conclusions

In this work, we introduced a framework to incorporate learnable diffusion into graph scattering network, allowing for data-driven feature extraction. The model is mathematically interpretable, with expressivity and stability guarantees maintained. We show that our approach expand the scenarios where scattering architectures are competitive to deep learning in term of performance, and reduce the performance gap between well-understood and non-transparent learning paradigms on larger data regimes, via graph classification experiments on social network and bioinformatics datasets.

Our results open up several promising research directions. One is to explore more designs of learnable operators beyond the self-adjoint constraint. Another is developing scattering-based models with interpretable later modules, akin to the Euclidean case, while integrating learnable diffusion into the scattering transform. For completeness, we note that preliminary experiments on much larger

data regimes, which are not discussed in detail here, indicate that the current scattering framework does not maintain a performance advantage compared to deep learning in those settings and is still outperformed by a significant, albeit narrowed, margin. We leave further investigation to future work.

Acknowledgments. Computing resources were sponsored by Intelligent Integration Co., Ltd. (INT2), Viet Nam. We thank the reviewers and meta-reviewer for their insightful suggestions which contributed to improving this paper.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article, beyond the computing resources provided by INT2, already acknowledged above.

References

1. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural computation **15**(6), 1373–1396 (2003)
2. Bronstein, A.M., Bronstein, M.M., Kimmel, R., Mahmoudi, M., Sapiro, G.: A gromov-hausdorff framework with diffusion geometry for topologically-robust non-rigid shape matching. International Journal of Computer Vision **89**(2), 266–286 (2010)
3. Bruna, J., Mallat, S.: Invariant scattering convolution networks. IEEE transactions on pattern analysis and machine intelligence **35**(8), 1872–1886 (2013)
4. Chamberlain, B., Rowbottom, J., Gorinova, M.I., Bronstein, M., Webb, S., Rossi, E.: Grand: Graph neural diffusion. In: International Conference on Machine Learning. pp. 1407–1418. PMLR (2021)
5. Coifman, R.R., Lafon, S.: Diffusion maps. Applied and computational harmonic analysis **21**(1), 5–30 (2006)
6. Coifman, R.R., Maggioni, M.: Diffusion wavelets. Applied and Computational Harmonic Analysis **21**(1), 53–94 (2006), special Issue: Diffusion Maps and Wavelets
7. Coifman, R., Lafon, S., Lee, A., Maggioni, M., Nadler, B., Warner, F., Zucker, S.: Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. PNAS **102**(21) (2005)
8. Coifman, R., Lafon, S., Lee, A., Maggioni, M., Nadler, B., Warner, F., Zucker, S.: Geometric diffusions as a tool for harmonic analysis and structure definition of data: Multiscale methods. PNAS **102**(21) (2005)
9. Gama, F., Ribeiro, A., Bruna, J.: Diffusion scattering transforms on graphs. In: International Conference on Learning Representations (2019)
10. Gama, F., Ribeiro, A., Bruna, J.: Stability of graph scattering transforms. Advances in Neural Information Processing Systems **32** (2019)
11. Gao, F., Wolf, G., Hirn, M.: Geometric scattering for graph data analysis. In: International Conference on Machine Learning. pp. 2122–2131. PMLR (2019)
12. Guth, F., Zarka, J., Mallat, S.: Phase Collapse in Neural Networks. In: International Conference on Learning Representations (2022)
13. Hammond, D.K., Vandergheynst, P., Gribonval, R.: Wavelets on graphs via spectral graph theory. Applied and Computational Harmonic Analysis **30**(2), 129–150 (2011)

14. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net (2017)
15. Mallat, S.: Group invariant scattering. Communications on Pure and Applied Mathematics **65**(10), 1331–1398 (2012)
16. Mallat, S.: A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way. Academic Press, Inc., USA, 3rd edn. (2008)
17. Morris, C., Kriege, N.M., Bause, F., Kersting, K., Mutzel, P., Neumann, M.: Tu-dataset: A collection of benchmark datasets for learning with graphs. In: ICML 2020 Workshop on Graph Representation Learning and Beyond (GRL+ 2020) (2020)
18. Nadler, B., Lafon, S., Kevrekidis, I., Coifman, R.: Diffusion maps, spectral clustering and eigenfunctions of fokker-planck operators. Advances in neural information processing systems **18** (2005)
19. Nguyen, D.Q., Nguyen, T.D., Phung, D.: Universal graph transformer self-attention networks. In: Companion Proceedings of the Web Conference 2022. pp. 193–196 (2022)
20. Oyallon, E., Zagoruyko, S., Huang, G., Komodakis, N., Lacoste-Julien, S., Blaschko, M., Belilovsky, E.: Scattering networks for hybrid representation learning. IEEE transactions on pattern analysis and machine intelligence **41**(9), 2208–2221 (2018)
21. Rudin, W.: Real and complex analysis, 3rd ed. McGraw-Hill, Inc., USA (1987)
22. Shuman, D.I., Wiesmeyr, C., Holighaus, N., Vandergheynst, P.: Spectrum-adapted tight graph wavelet and vertex-frequency frames. IEEE Transactions on Signal Processing **63**(16), 4223–4235 (2015)
23. Sun, C., Hu, J., Gu, H., Chen, J., Yang, M.: Adaptive graph diffusion networks (2022), https://arxiv.org/abs/2012.15024
24. Tong, A., Wenkel, F., Bhaskar, D., Macdonald, K., Grady, J., Perlmutter, M., Krishnaswamy, S., Wolf, G.: Learnable filters for geometric scattering modules. IEEE Transactions on Signal Processing (2024)
25. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017)
26. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph Attention Networks. International Conference on Learning Representations (2018)
27. Wang, Y., Wang, Y., Yang, J., Lin, Z.: Dissecting the diffusion process in linear graph convolutional networks. Advances in Neural Information Processing Systems **34**, 5758–5769 (2021)
28. Weiss, Y.: Segmentation using eigenvectors: a unifying view. In: Proceedings of the Seventh IEEE International Conference on Computer Vision. vol. 2, pp. 975–982 vol.2 (1999). https://doi.org/10.1109/ICCV.1999.790354
29. Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? In: International Conference on Learning Representations (2018)
30. Zarka, J., Guth, F., Mallat, S.: Separation and concentration in deep networks. In: ICLR 2021-9th International Conference on Learning Representations (2021)
31. Zarka, J., Thiry, L., Angles, T., Mallat, S.: Deep network classification by scattering and homotopy dictionary learning. In: ICLR 2020-8th International Conference on Learning Representations (2020)

32. Zhao, J., Dong, Y., Ding, M., Kharlamov, E., Tang, J.: Adaptive diffusion in graph neural networks. Advances in neural information processing systems **34**, 23321–23333 (2021)
33. Zou, D., Lerman, G.: Graph convolutional neural networks via scattering. arXiv preprint arXiv:1804.00099 (2018)