BLIS-Net: Classifying and Analyzing Signals on Graphs

Charles Xu
Yale University

Laney Goldman
Harvey Mudd College

Valentina Guo
Yale University

Benjamin Hollander-Bodie
Yale University

Maedee Trank-Greene
University of Colorado Boulder

Ian Adelstein
Yale University

Edward De Brouwer
Yale University

Rex Ying
Yale University

Smita Krishnaswamy
Yale University

Michael Perlmutter
Boise State University

Abstract

Graph neural networks (GNNs) have emerged as a powerful tool for tasks such as node classification and graph classification. However, much less work has been done on signal classification, where the data consists of many functions (referred to as signals) defined on the vertices of a single graph. These tasks require networks designed differently from those designed for traditional GNN tasks. Indeed, traditional GNNs rely on localized low-pass filters, and signals of interest may have intricate multi-frequency behavior and exhibit long range interactions. This motivates us to introduce the BLIS-Net (Bi-Lipschitz Scattering Net), a novel GNN that builds on the previously introduced geometric scattering transform. Our network is able to capture both local and global signal structure and is able to capture both low-frequency and high-frequency information. We make several crucial changes to the original geometric scattering architecture which we prove increase the ability of our network to capture information about the input signal and show that BLIS-Net achieves superior performance on both synthetic and real-world data sets based on traffic flow and fMRI data.

Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238. Copyright 2024 by the author(s). Correspondence: smita.krishnaswamy@yale.edu.

1 INTRODUCTION

Recent years have seen a tremendous rise of Graph Neural Networks (GNNs) as a powerful tool for processing graph-structured data (Wu et al., 2020). Most of the research has focused on three families of tasks: graph-level tasks in which the data consists of many different graphs (Errica et al., 2019), node-level tasks (Kipf and Welling, 2016) such as classifying each user in a large network, and edge-level tasks such as link prediction (Zhang and Chen, 2018). However, there is another family of problems, signal-level tasks, which has received comparatively little attention.

Here we focus on developing a high-performing network for signal classification, where the goal is to predict the label y_i of a signal (function) $\mathbf{x_i} : V \to \mathbb{R}$ defined on the vertices of a weighted graph $G = (V, E, w)$. Notably, this is the natural generalization of image classification, which can be thought of as classifying many signals defined on a grid graph, to more irregular domains.

Additionally, we note that signal-level tasks have many practical applications. For example, in traffic data, the road network is kept fixed but the number of cars at each intersection varies each day. A natural goal would be to identify, and then analyze, anomalous traffic patterns. In the analysis of brain-scan data, a patient's neuronal structures (i.e., voxels, neurons) can be modeled as a fixed graph with different levels of activity in each region across time, offering a useful framework for analyzing neural data (Li et al., 2021).

However, GNNs that have been designed with node-level tasks in mind perform limited processing from a signal perspective. A foundational principle of most node-level analysis is homophily (Zhu et al., 2020), the idea that nodes are similar to their neighbors. Therefore, one wants to produce a hidden representation of each node which varies slowly among neighbors. However, when focused on signal-level tasks, the local homophily heuristic is not applicable and we find that it is important to capture (i) both low-frequency and high-frequency information in the input signal and (ii) both local and global behavior.

A natural choice when working with signals is to use methods from graph signal processing (Shuman et al., 2013) incorporated into a neural network. To this end, we rely on the geometric scattering transform (Gao et al., 2019; Gama et al., 2019b,a; Zou and Lerman, 2019b), a multi-order, multi-scale transform that alternates wavelet transforms and non-linear modulus activations in the form of a deep (although typically fixed) network. The wavelets act as band-pass filters that capture information at different frequencies and scales. Therefore, geometric scattering provides a solid starting place for signal-level tasks.

However, geometric scattering alone is insufficient for two key reasons: First, we establish that geometric scattering is not injective due to its modulus operation, and thus loses expressivity since it cannot distinguish between certain signal classes. Second, scattering produces an unnecessarily high-dimensional representation of the signal that is not tuned to classification purposes. This motivates us to introduce BLIS-Net (Bi-Lipschitz Scattering Net) which incorporates advances to address these issues while facilitating integration into larger neural networks.

BLIS-Net builds on previous work on the geometric scattering transform and introduces ReLU and reflected ReLU activations to preserve injectivity. Further, BLIS-Net incorporates dimension reduction and classification modules to demonstrate the modular use of bi-Lipschitz Scattering within a larger ML context. Our contributions can be summarized as follows:

1. We introduce BLIS-Net, a novel GNN for signal classification on graphs.
2. We prove two theorems (Theorem 3.1 and Theorem 3.2), which, when considered jointly, show that the BLIS module is provably more expressive than the geometric scattering transform. In particular, Theorem 3.2 shows that BLIS is bi-Lipschitz and therefore stably invertible.
3. We show that BLIS-Net achieves superior performance on both synthetic data and real-world data sets derived from traffic and fMRI data.

2 BACKGROUND

2.1 Notation and Preliminaries

Let $G = (V, E, w)$ be a weighted, connected graph with vertices $V = \{v_1, \cdots, v_n\}$. Throughout this paper, we will consider functions $\mathbf{x} : V \to \mathbb{R}$, which we refer to as *graph signals*, and will in a slight abuse of notation not distinguish between the signal $\mathbf{x}$ and the vector $\mathbf{x} \in \mathbb{R}^n$ defined by $x_i = \mathbf{x}(v_i)$. We will let A be the weighted adjacency of G, let $\mathbf{d} = A\mathbb{1}$ denote the weighted degree vector, and let $D = \text{diag}(\mathbf{d})$ be the degree matrix.

We will let $L_N = I - D^{-1/2}AD^{-1/2}$ denote the symmetric normalized graph Laplacian. It is well-known that L_N is positive semidefinite and admits an orthonormal basis (ONB) of eigenvectors with $L_N\mathbf{v_i} = \omega_i\mathbf{v_i}$ with $0 = \omega_1 < \omega_2 \leq \ldots \leq \omega_n \leq 2$ (where the fact that $\omega_2 > 0$ follows from the assumption that G is connected). This allows us to write $L_N = V\Omega V^T$, where V is a matrix whose i-th column is $\mathbf{v_i}$ and Ω is a diagonal matrix with $\Omega_{i,i} = \omega_i$. Since the $\{\mathbf{v_i}\}_{i=1}^n$ form an ONB, we see that V is unitary and $V^TV = I$.

It is known (e.g., Section 2 of Min et al. (2022)) that

$$\mathbf{x}^T L_N \mathbf{x} = \sum_{\{v_i,v_j\}\in E} (\tilde{x}_i - \tilde{x}_j)^2 \tag{1}$$

where $\tilde{\mathbf{x}} \coloneqq \mathbf{D}^{-1/2}\mathbf{x}$ is a normalized version of $\mathbf{x}$. Therefore L_N is viewed as a matrix whose quadratic form measures the smoothness of (normalized) signals. If we take $\mathbf{x} = \mathbf{v_i}$ we have $\mathbf{v_i}^T L_N \mathbf{v_i} = \omega_i$. Therefore, we may interpret each eigenvalue ω_i as a frequency and each eigenvector as a generalized Fourier mode. The high-frequency eigenvectors oscillate rapidly within local neighborhoods leading to large values in (1) whereas the low-frequency eigenvectors are smooth in the sense that they vary slowly within graph neighborhoods. Therefore, we view methods based on the eigendecomposition of the graph Laplacian as the natural extension of traditional signal processing to the graph setting.

We note that since V is unitary, we have $p(L_N) = Vp(\Omega)V^T$ for any polynomial p. Thus, for any continuous function $f : [0, 2] \to \mathbb{R}$ and diagonalizable matrix $M = B\Xi B^{-1}$ with $\Xi = \text{diag}(\xi_1, \ldots, \xi_n)$, we define

$$f(M) = Bf(\Xi)B^{-1}, \tag{2}$$

where $f(\Xi) = \text{diag}(f(\xi_1), \ldots, f(\xi_n))$.

2.2 Graph signals and signal-level tasks

Graph signal-level tasks naturally arise in biological, natural, and social systems. Key examples include:

- Predicting properties of social networks. For instance, while classifying the political affiliation of an individual is a node-level task, characterizing a polarized population (low-frequency) vs a non-polarized population (high-frequency) is a signal-level task.

- Networks that occur in nature such as cell-communication networks have genes or cytokines as signals on a fixed graph substrate (Moon et al., 2019). In many of these cases, the number of signals on the network is close to the number of nodes.
- In neuroscience, one can represent brain measurements as signals living on a fixed graph. The graph embodies the connectivity between different brain regions and the signal would be the brain activity measurements in each brain region. A typical task is then to predict external stimuli from brain signals (Ménoret et al., 2017).

In general, a signal-level task is any machine learning task, e.g., classification, regression, or clustering, where the data set consists of many different signals defined on a single fixed graph.

2.3 Diffusion Matrices

Let $g(t)$ be a decreasing function on $[0, 2]$ with $g(0) = 1, g(2) = 0$, and let $T = g\left(L_N\right)$ (defined as in (2)). By construction, T is diagonalizable and $T = V \Lambda V^T$, where $\Lambda := g\left(\Omega\right)$. As our primary example, which we will use in all of our numerical experiments, we will let

$$g(t) = 1 - \frac{t}{2}. \tag{3}$$

T then becomes the symmetrized diffusion operator

$$T = I - \frac{L_N}{2} = \frac{1}{2}\left(I + D^{-1/2} A D^{-1/2}\right).$$

Next, we let $\mathbf{w} \in \mathbb{R}^n$ be a weight vector with $w_i > 0$, let $W := \text{diag}(\mathbf{w})$, and K the asymmetric diffusion matrix

$$K := W^{-1} T W. \tag{4}$$

We let $\mathbf{L}^2_{\mathbf{w}}$ be the weighted inner product space with $\langle \mathbf{x}, \mathbf{y} \rangle_{\mathbf{w}} = \langle W\mathbf{x}, W\mathbf{y} \rangle_2 = \sum_{i=1}^n x_i y_i w_i^2$, and norm denoted by $\|\cdot\|_{\mathbf{w}}$. One may verify that K is self-adjoint on $\mathbf{L}^2_{\mathbf{w}}$ (see Perlmutter et al. (2023), Lemma 1.1).

We note that if we set $W = D^{-1/2}$, then K becomes the lazy random walk matrix, $P := \frac{1}{2}\left(I + AD^{-1}\right)$, which was used to construct diffusion wavelets in Gao et al. (2019), whereas if we set $W = I$, K is simply equal to the matrix T which was used in Gama et al. (2019b). More generally, Perlmutter et al. (2023) considered $W = D^\alpha$, $-.5 \leq \alpha \leq .5$ and found empirically that the optimal choice of α varied from task to task.

2.4 Graph Wavelets and Frame Properties

Let $J \geq 0$, and let $\mathcal{F} = \{F_j\}_{j=0}^{J+1}$ be a collection of $n \times n$ matrices. We say that $\mathcal{F}$ is a *frame* on $\mathbf{L}^2_{\mathbf{w}}$ if there exist constants $0 < c \leq C < \infty$ such that,

$$c\|\mathbf{x}\|^2_{\mathbf{w}} \leq \sum_{j=0}^{J+1} \|F_j \mathbf{x}\|^2_{\mathbf{w}} \leq C\|\mathbf{x}\|^2_{\mathbf{w}}, \quad \forall \mathbf{x} \in \mathbb{R}^n. \tag{5}$$

We say that $\mathcal{F}$ is a *non-expansive frame* if $C \leq 1$ and that $\mathcal{F}$ is an *isometry* if $c = C = 1$.

We now construct two families of wavelet frames. Analogous to Tong et al. (2022), let $\{s_j\}_{j=0}^{J+1}$ be a sequence of scales with $s_0 = 0$ and $s_1 = 1$, and $s_j < s_{j+1}$. For $0 \leq j \leq J$, let $p_j(t)$ denote the polynomial defined by $p_j(t) := t^{s_j} - t^{s_{j+1}}$ and let $p_{J+1} := t^{s_{J+1}}$. By construction, each p_j is nonnegative on $[0, 1]$, and therefore, we may define $q_j(t) := p_j(t)^{1/2}$ for $0 \leq t \leq 1$.

We then define two graph wavelet transforms $\mathcal{W}^{(1)}_J := \{\Psi^{(1)}_j, \Phi^{(1)}_J\}_{0 \leq j \leq J}$, where $\Psi^{(1)}_j := q_j(K)$, $\Phi^{(1)}_J := q_{J+1}(K)$ (defined as in (2)) and $\mathcal{W}^{(2)}_J := \{\Psi^{(2)}_j, \Phi^{(2)}_J\}_{0 \leq j \leq J}$, $\Psi^{(2)}_j := p_j(K)$, $\Phi^{(2)}_J := p_{J+1}(K)$.

The following result shows that $\mathcal{W}^{(1)}_J$ is an isometry and that $\mathcal{W}^{(2)}_J$ is a non-expansive frame on $\mathbf{L}^2_{\mathbf{w}}$.[1]

Proposition 2.1. *For any $\mathbf{x} \in \mathbb{R}^n$, we have,*

$$\|\mathcal{W}^{(1)}_J \mathbf{x}\|^2_{\mathbf{w}} := \|\Phi^{(1)}_J \mathbf{x}\|^2_{\mathbf{w}} + \sum_{j=0}^{J} \|\Psi^{(1)}_j \mathbf{x}\|^2_{\mathbf{w}} = \|\mathbf{x}\|^2_{\mathbf{w}}. \tag{6}$$

Additionally, there exists a constant $c > 0$, depending only on the maximal scale s_{J+1}, such that

$$c\|\mathbf{x}\|^2_{\mathbf{w}} \leq \|\mathcal{W}^{(2)}_J \mathbf{x}\|^2_{\mathbf{w}} \leq \|\mathbf{x}\|^2_{\mathbf{w}} \quad \textit{for all } \mathbf{x} \in \mathbb{R}^n, \tag{7}$$

where $\|\mathcal{W}^{(2)}_J \mathbf{x}\|^2_{\mathbf{w}}$ is defined analogously to $\|\mathcal{W}^{(1)}_J \mathbf{x}\|^2_{\mathbf{w}}$.

2.5 The Graph Scattering Transform

Given a wavelet frame $\mathcal{W}_J = \{\Psi_j\}_{j=0}^J \cup \{\Phi_J\}$ such as $\mathcal{W}^{(1)}_J$ and $\mathcal{W}^{(2)}_J$, the graph scattering transform is a multilayer feedforward network consisting of alternating wavelet transforms and non-linearities (building off of an analogous construction (Mallat, 2012) modeling CNNs for Euclidean data such as images). In particular, given a sequence of scales $j_1, \ldots, j_m$, we define

$$U[j_1, \ldots, j_m]\mathbf{x} = H\Psi_{j_m} \ldots H\Psi_{j_1}\mathbf{x}, \tag{8}$$

where $H\mathbf{x} = |\mathbf{x}|$ is the componentwise modulus (absolute value) operator $(H\mathbf{x})_i = |x_i|$. Then, after computing each of the $U[j_1, \ldots, j_m]\mathbf{x}$, one may extract m-th

[1] Full proofs of all theoretical results are provided in the supplementary materials.

order scattering coefficients via the low-pass filter Φ_J,

$$S_J[j_1,\ldots,j_m]\mathbf{x} = \Phi_J U[j_1,\ldots,j_m]\mathbf{x}.$$

If one wishes to apply the graph scattering transform to tasks such as node classification, they may compute the scattering coefficients up to order M and take the scattering coefficients evaluated at each vertex, $\{S_J[j_1,\ldots,j_m]\mathbf{x}(v) : 0 \leq m \leq M, 0 \leq j_1,\ldots,j_m \leq J\}$ as a collection of node features which may then be input into another machine learning algorithm such as a multilayer perceptron. Alternatively, if one wishes to apply the graph scattering transform to whole-graph level tasks such as graph classification, one first performs a global aggregation such as summation or moment aggregation (Gao et al., 2019) before applying the final classifier. Importantly, we note that coefficients of orders $m = 0,\ldots,M$ (where the zeroth-order coefficient is simply $\Phi_J\mathbf{x}$) are fed into the classifier, not just the m-th order scattering coefficients.

3 BLIS-Net

Here, we introduce BLIS-Net, a novel neural network for graph signals which, as discussed in Section 2.2, arise frequently in the natural and behavioral sciences.

In order to create a network that can classify or regress properties of signals, one needs to create a rich representation of the signal. A natural choice for such a representation is a signal processing transform such as the geometric scattering transform discussed in Section 2.5. Indeed, it was shown that the geometric scattering transform was an effective tool for identifying anomalies in traffic data in Bodmann and Emilsdottir (2022). However, the geometric scattering transform has limitations in its ability to characterize its input signal, which motivates us to introduce the BLIS module.

The primary deficiency of the geometric scattering transform which we seek to address is lack of injectivity. Since the scattering transform is constructed using the modulus in (8), it is trivial that the scattering transform will produce identical representations of $\mathbf{x}$ and $-\mathbf{x}$. The following theorem shows that there are also non-trivial examples of distinct signals with identical scattering coefficients. This may be proved by constructing signals $\mathbf{x_1}$ and $\mathbf{x_2}$, where each $\mathbf{x_j}$ is supported on two disjoint regions, such that $\mathbf{x_1} \neq \pm\mathbf{x_2}$, but $\mathbf{x_1}$ and $\mathbf{x_2}$ have identical scattering coefficients. We also note that in Section 4.1, we conduct experiments on synthetic data modeled after this choice of $\mathbf{x_1}$ and $\mathbf{x_2}$ to further illustrate the limitations of the geometric scattering transform which are addressed by BLIS.

Theorem 3.1. *There exist signals $\mathbf{x_1}$ and $\mathbf{x_2}$ with identical scattering coefficients such that $\mathbf{x_1} \neq \pm\mathbf{x_2}$*[2].

This result shows that the geometric scattering transform has limits on its expressive power since there are non-equivalent signals of which it produces identical representations. Notably, the importance of injectivity has also been noted in the context of graph classification. In particular, Xu et al. (2019) showed that using an injective aggregation function (in a message-passing network) was the key to producing a maximally expressive graph neural network. We also note that the result of Theorem 3.1 is somewhat surprising since Mallat and Waldspurger (2015) showed there were no non-trivial signal pairs with identical scattering coefficients for the original Euclidean scattering transform (Mallat, 2012).

3.1 The BLIS Module

Recall from Section 2.5 that the geometric scattering transform uses an alternating sequence of wavelet transforms and componentwise modulus operators to produce coefficients such as $S_J[j_1,j_2]\mathbf{x} = \Phi_J H\Psi_{j_2} H\Psi_{j_1}\mathbf{x}$. BLIS makes the following modifications:

1. To induce injectivity, BLIS uses two different activation functions $\sigma_1(\mathbf{x}) := \mathrm{ReLU}(\mathbf{x})$ and $\sigma_2(\mathbf{x}) := \mathrm{ReLU}(-\mathbf{x})$. Notably, we have

   $$\sigma_1(\mathbf{x}) + \sigma_2(\mathbf{x}) = M\mathbf{x}.$$

   Thus, the use of σ_1 and σ_2 may be viewed as decomposing the absolute value into two disjointly supported non-linearities. Additionally, this modification is crucial to proving Theorem 3.2 which shows that the BLIS module is injective on $\mathbb{R}^n$.

2. To account for all frequency bands of the signal, BLIS uses the entire wavelet frame $\mathcal{W}_J = \{\Psi_j\}_{j=0}^J \cup \{\Phi_J\}$ in each layer. This is in contrast to the geometric scattering transform which does not utilize the low-pass filter Φ_J until after the final non-linearity. This modification is needed to ensure that BLIS has the bi-Lipshitz property established in Theorem 3.2 and is also key to the conservation of energy property established in Theorem 3.4. This latter property ensures that BLIS doesn't lose information which may be critical for tasks such as classification.

3. Since BLIS uses the entire wavelet frame in each layer, all of the energy of the input signal is preserved in each layer. Therefore, the only output of an m-layer BLIS module is the coefficients produced in the final layer (i.e., through a sequence of m filterings followed by activations). This is in contrast to the geometric scattering transform which outputs first-order coefficients $S_J[j_1]\mathbf{x}$, second-order coefficients $S_J[j_1,j_2]\mathbf{x}$, etc., up to m-th order coefficients $S_J[j_1,\ldots,j_m]\mathbf{x}$ (in

[2] See the supplement for a detailed theorem statement.

addition to a single zeroth-order coefficient which is simply $\Phi_J\mathbf{x}$). This makes it straightforward to incorporate the BLIS module into a neural network without the need for skip connections.

To explicitly define the BLIS module, we rewrite the wavelet frame $\mathcal{W}_J = \{\Psi_j\}_{j=0}^{J} \cup \{\Phi_J\}$ as $\mathcal{F} = \{F_j\}_{j=0}^{J+1}$ where $F_j = \Psi_j$ for $0 \leq j \leq J$ and $F_{J+1} = \Phi_J$. We let $m \geq 1$ denote the depth of the network and define

$$\begin{aligned} &B[j_1, k_1, \cdots, j_m, k_m](\mathbf{x}) \\ :=&\sigma_{k_m}(F_{j_m}\sigma_{k_{m-1}}(F_{j_{m-1}} \cdots \sigma_{k_1}(F_{j_1}\mathbf{x}))\cdots) \end{aligned} \tag{9}$$

for $0 \leq j_i \leq J+1$, and $k_i \in \{1, 2\}$. We then let $\mathbf{B}_m(\mathbf{x})$ denote the set of all the $B[j_1, k_1, \ldots, j_m, k_m]$.

We remark that one could readily modify the BLIS framework to include other frames $\mathcal{F}$ in place of the wavelets $\mathcal{W}_J^{(1)}$ or $\mathcal{W}_J^{(2)}$. For example, one could use the spectral wavelets considered in Zou and Lerman (2019b) or frames obtained as the union of different wavelet families. Importantly, the proofs of Theorems 3.2 and 3.4 do not depend on the specific wavelet construction, but only on the frame constants $0 < c \leq C < \infty$. Therefore, both of these results apply to variations of BLIS constructed via arbitrary $\mathcal{F}$ satisfying (5).

3.2 The bi-Lipschitz Property

Theorem 3.1, stated above, shows that the geometric scattering transform is not injective on $\mathbb{R}^n$ (even up to the equivalence relation $\mathbf{x} \sim \pm\mathbf{x}$). Therefore, it may lack the ability to effectively characterize graph signals. By contrast, the following theorem shows that BLIS is a bi-Lipschitz map on weighted inner product space $\mathbf{L}^2_{\mathbf{w}}$ introduced in Section 2.3 where we equip the image space with the mixed norm obtained by taking the (unweighted) ℓ^2 norm of the weighted ℓ^2 norms of the individual $B[j_1, k_1, \ldots, j_m, k_m]\mathbf{x}$, so that

$$\|\mathbf{B}_m(\mathbf{x})\|^2_{\mathbf{w},2} = \sum_{k_i=1}^{2} \sum_{j_i=0}^{J+1} \|B[j_1, k_1, \cdots, j_m, k_m](\mathbf{x})\|^2_{\mathbf{w}}.$$

Theorem 3.2. *$\mathbf{B}_m$ is bi-Lipshitz on $\mathbf{L}^2_{\mathbf{w}}$, i.e.,*

$$\left(\frac{c}{2}\right)^m \|\mathbf{x}-\mathbf{y}\|^2_{\mathbf{w}} \leq \|\mathbf{B}_m(\mathbf{x}) - \mathbf{B}_m(\mathbf{y})\|^2_{\mathbf{w},2} \leq C^m \|\mathbf{x}-\mathbf{y}\|^2_{\mathbf{w}}$$

for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, where $0 < c \leq C < \infty$ are the frame bounds for the wavelets defined as in (5).

The following corollary is immediate from the first inequality in Theorem 3.2 and the definition of a norm.

Corollary 3.1. *$\mathbf{B}_m$ is injective on $\mathbb{R}^n$.*

We also note that the lower bound established in Theorem 3.2 implies the existence of a Lipschitz continuous inverse map[3] that reconstructs $\mathbf{x}$ from $\mathbf{B}_m(\mathbf{x})$. This property is particularly interesting in light of work (Zou and Lerman, 2019a; Castro et al., 2020; Bhaskar et al., 2022) which has attempted to invert the geometric scattering transform as part of an encoder-decoder graph-generation network.

3.3 Properties inherited from scattering

In addition to the bi-Lipschitz property, we may also show that BLIS retains desirable theoretical properties from the geometric scattering transform such as permutation equivariance and conservation of energy.

Permutation equivariance is the property that if we reorder the vertices $v_1, \ldots, v_n$ (and therefore reorder the entries of the input signal since $x_i = \mathbf{x}(v_i)$), then the representations of the vertices are reordered in the same manner. It is crucial to the success of a well-designed GNN since it ensures that the network captures the intrinsic graph structure of the data rather than relying on the ordering of the vertices. The following theorem shows BLIS is permutation equivariant.

Theorem 3.3. *Let Π be a permutation matrix corresponding to a reordering of the nodes. Then,*

$$\Pi B[j_1, k_1, \cdots, j_m, k_m]\mathbf{x} = B[j_1, k_1, \cdots, j_m, k_m]\Pi\mathbf{x},$$

for all $j_1, k_1, \ldots, j_m, k_m$, where on the right-hand side $B[j_1, k_1, \ldots, j_m, k_m]$ is defined in terms of the permuted ordering (with the permuted weight vector $\Pi\mathbf{w}$).

In our aggregation module (discussed below in Section 3.4), we perform a global summation over the vertices. In light of Theorem 3.3, we sum the same terms on both the original and the permuted graph, just in a different order. Therefore, the output of the aggregation module is the same for both graphs. Thus, the BLIS module produces an equivariant representation of the signal from which the aggregation module extracts invariance.

Previous work has shown that the infinite-depth geometric scattering transform preserves the norm of the input. For example, Theorem 3.5 of Perlmutter et al. (2023) shows that if the scattering transform is constructed using $\mathcal{W}_J^{(1)}$, then

$$\sum_{m=0}^{\infty} \sum_{0 \leq j_i \leq J} \|S_J[j_1, \ldots, j_m]\mathbf{x}\|^2_{\mathbf{w}} = \|\mathbf{x}\|^2_{\mathbf{w}}.$$

We may derive an analogous result for BLIS. However, our theorem differs from previous work in that it shows that the energy of the input signal is preserved in each layer (whereas previous work showed energy was conserved when summing over all layers). Indeed, this

[3] Discussed further in the supplement.

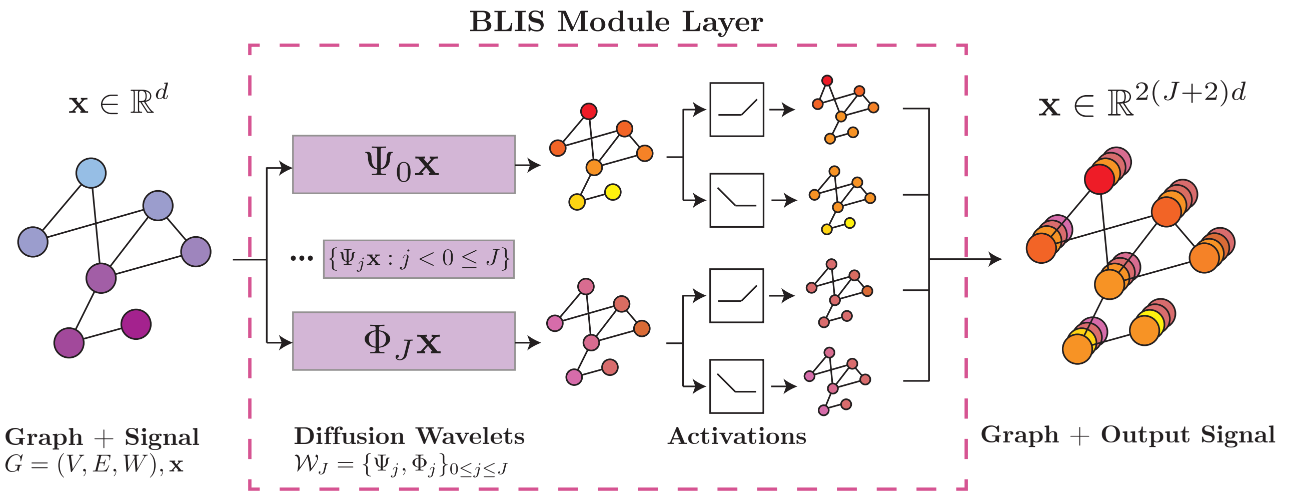
BLIS Module Layer
$\mathbf{x} \in \mathbb{R}^d$
$\Psi_0 \mathbf{x}$
$\{\Psi_j \mathbf{x} : j < 0 \leq J\}$
$\Phi_J \mathbf{x}$
$\mathbf{x} \in \mathbb{R}^{2(J+2)d}$
Graph + Signal
$G = (V, E, W), \mathbf{x}$
Diffusion Wavelets
$\mathcal{W}_J = \{\Psi_j, \Phi_j\}_{0 \leq j \leq J}$
Activations
Graph + Output Signal

Figure 1: The BLIS module: We first apply multiscale diffusion wavelet transform to the input signal and then two activation functions, σ_1 and σ_2. The output is a multivariate signal, with $2(J+2)$ times the original dimension.

result helps motivate the third modification discussed at the beginning of Section 3.1, where we implement BLIS without skip connections, unlike scattering.

Theorem 3.4. *For all* $\mathbf{x} \in \mathbb{R}^n$, *we have*

$$c^m \|\mathbf{x}\|_{\mathbf{w}}^2 \leq \|\mathbf{B}_m(\mathbf{x})\|_{\mathbf{w},2}^2 \leq C^m \|\mathbf{x}\|_{\mathbf{w}}^2 ,$$

where $0 < c \leq C < \infty$ *are the frame bounds for the wavelets defined as in* (5). *In particular, if* $c = C = 1$, *as is the case for* $\mathcal{W}_J^{(1)}$, *we have* $\|\mathbf{B}_m(\mathbf{x})\|_{\mathbf{w},2} = \|\mathbf{x}\|_{\mathbf{w}}$.

3.4 BLIS-Net Architecture

The bi-Lipschitz Scattering Network (BLIS-Net) integrates the BLIS module with several other modules:

1. **BLIS module layer**: The first layer of our network utilizes the BLIS module to extract features from the input graph signal.
2. **Moment aggregation module**: For each $j_1, k_1, \ldots, j_m, k_m$, we aggregate the BLIS features across the nodes, i.e.,

$$\begin{aligned} &B'[j_1, k_1, \cdots, j_m, k_m]\mathbf{x_i} \\ &= \sum_{v \in V} B'[j_1, k_1, \cdots, j_m, k_m](\mathbf{x_i})(v). \end{aligned}$$

3. **Embedding layer**: To reduce the risk of overfitting and increase computational efficiency, we next perform a dimensionality reduction via an embedding layer.
4. **Classification layer**: Finally we include an MLP classifier featuring softmax activation.

We note that although the focus of this paper is signal classification, other common machine learning tasks such as clustering and regression have natural analogs in the signal setting. Our method can be flexibly adapted to these tasks due to its modular design. For example, one could train a clustering algorithm on top of the output of the BLIS module.

4 EXPERIMENTAL RESULTS

We demonstrate the utility of BLIS-Net (with both $\mathcal{W}_J^{(1)}$ and $\mathcal{W}_J^{(2)}$) on synthetic and real-world data sets. As baselines, we use several widely adopted graph neural networks based on several variations of the message-passing framework: the Graph Convolutional Network (GCN) (Kipf and Welling, 2016), Graph Attention Network (GAT) (Veličković et al., 2018), and the Chebyshev spectral graph convolutional operator from ChebNet (Defferrard et al., 2016). We also consider several networks with powerful graph distinction abilities, such as the Graph Isomorphism Network (GIN) (Xu et al., 2019), GNNML1 and GNNML3 (Balcilar et al., 2021), and Provably Powerful Graph Networks (PPGN) (Maron et al., 2019). We also consider the general, powerful, scalable (GPS) graph transformer (Rampášek et al., 2022) which has achieved state-of-the-art performance on a wide range of benchmarks. We note that, unlike message-passing GNNs, the GPS allows information to spread across the graph via full connectivity, thus allowing the network to capture global properties of the signal. We additionally compare against the geometric scattering transform (Gao et al., 2019), which is simply labeled as Scattering in our tables. In our tables, we color the top-performing and second-best method. Further details on model implementation, computational complexity, data sets, hyperparameters, and training procedures are provided in the supplement.

4.1 Synthetic data

We first generate $2N$ random functions $f_1^{(1)}, \dots, f_N^{(1)}$ and $f_1^{(2)}, \dots, f_N^{(2)}$ defined on $[0,1]^2$ from two different distributions. We then define graph signals $\mathbf{x_k^{(j)}}$ with $(x_k^{(j)})_i = f_k^{(j)}(v_i)$ where the vertices $v_1, \dots, v_n$ are chosen-uniformly at random from $[0,1]^2$ and connected to their k-nearest neighbors. Our goal is to predict which distribution the signal was generated from.

In particular, we consider two families of functions

$$f_j^{(1)} = g_{\mu_1,\sigma_1} + g_{\mu_2,\sigma_2}, \quad f_j^{(2)} = g_{\mu_1,\sigma_1} - g_{\mu_2,\sigma_2},$$

where $g_{\mu,\sigma}(x) := \exp\left(-\frac{\|x-\mu_1^j\|_2^2}{2(\sigma_1^j)^2}\right)$ is a Gaussian function with center μ and bandwidth σ. In our first set of experiments, we generate values of μ_1^j and μ_2^j uniformly at random from $[0,1]^2$, set $\sigma_1^j = \sigma_2^j = \sigma^j$, where σ_j is chosen uniformly from $[0,1]$. Our second setup is similar, but with $\mu_1^j = \mu_2^j$, $\sigma_2 = \sigma_1/2$. Results of all methods are presented in Table 4.1.

The first setting, $\mu_1 \neq \mu_2$, results in signals modeled after those used in the proof of Theorem 3.1 with support concentrated near two, possibly far away, points. Therefore, we unsurprisingly observe that BLIS, as well as GAT, GIN, GCN, and GPS, perform well on this task whereas scattering is the least accurate method, likely due to its use of the absolute value.[4]

In our second setting, $\mu_1 = \mu_2 = \mu$ and $\sigma_2 = \sigma_1/2$, we view the Gaussians as two signals interfering with each other. Unlike the first setup, the absolute value does not severely limit the ability of the scattering transform to distinguish the signal classes. Indeed, the primary difference between these two signal classes is an oscillatory pattern in the middle of the signals' support. We observe that the two wavelet-based methods (Scattering and BLIS) are well equipped to capture this signal oscillation and both outperform GIN, GAT, and GCN (with BLIS outperforming scattering due to its increased expressive power). The utility of wavelets for this task is visualized in Figure 3, where we show that the two signal classes have markedly different responses to wavelet filters, but comparatively similar responses to the low-pass filter used in GCN. Further details on our experimental setup as well as ablation studies, where we also consider the geometric scattering transform and the BLIS module paired with shallow classifiers, e.g., SVM, are presented in the supplement.

Synthetic	Different μ	Same μ
GCN	99.0 ± 0.4	91.7 ± 2.0
GAT	98.6 ± 0.8	96.4 ± 0.6
GIN	**99.5 ± 0.2**	91.3 ± 1.4
GPS	95.4 ± 5.9	**97.7 ± 0.9**
ChebNet	98.8 ± 0.3	97.3 ± 0.2
GNNML1	99.3 ± 0.2	98.3 ± 0.4
GNNML3	99.8 ± 0.0	98.8 ± 0.4
PPGN	77.7 ± 9.9	62.4 ± 6.0
Scattering (W1)	97.7 ± 1.0	96.5 ± 1.2
Scattering (W2)	88.3 ± 4.3	96.8 ± 1.0
BLIS-Net (W1)	**100.0 ± 0.0**	**97.7 ± 0.5**
BLIS-Net (W2)	**99.5 ± 0.3**	**98.6 ± 0.4**

Table 1: Accuracy on the synthetic data sets. For BLIS-Net and Scattering, we consider the utilization of both wavelet families (denoted in parentheses as either W1 or W2) with dyadic scale sequences.

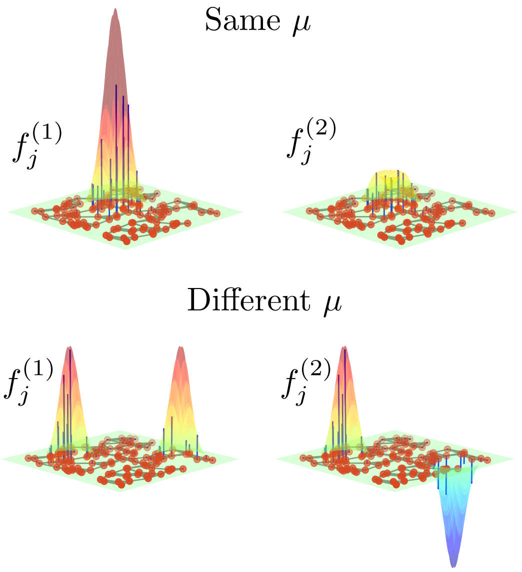

Figure 2: Synthetic signals $f_j^{(1)}$ and $f_j^{(2)}$.

4.2 Caltrans Traffic data

Here we consider data consisting of highway traffic measurements collected by the Caltrans Performance Measurement System (PeMS) (Chen et al., 2001), where over 39,000 sensors are deployed across California highways and data are aggregated every five minutes. PeMS03 and PeMS07 consist of traffic data

[4] Code needed to reproduce our experiments is available at `https://github.com/KrishnaswamyLab/blis`. Experiments were performed on a computing cluster with 8 CPUs and 4 NVIDIA RTX 5000 GPUs.

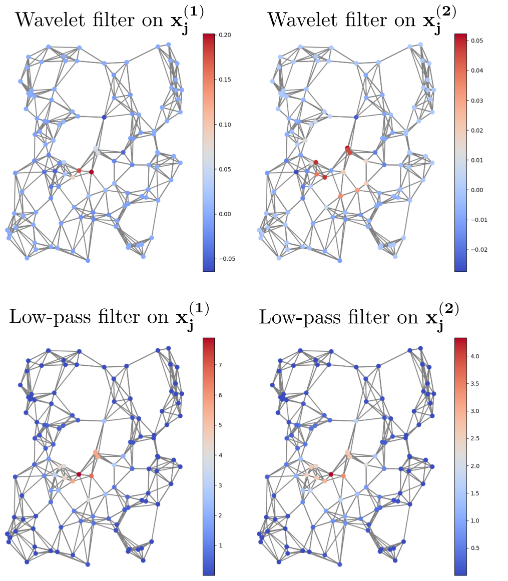

Figure 3: Filter responses on $\mathbf{x_j^{(1)}}$ and $\mathbf{x_j^{(2)}}$.

PEMS03	HOUR	DAY	WEEK
GCN	27.8 ± 2.0	14.1 ± 0.1	30.8 ± 0.5
GAT	36.5 ± 1.1	23.5 ± 0.8	30.8 ± 0.5
GIN	14.0 ± 12.4	14.3 ± 0.4	30.8 ± 0.5
GPS	57.4 ± 0.1	49.6 ± 0.3	31.9 ± 0.4
ChebNet	58.8 ± 1.6	56.8 ± 1.4	61.9 ± 3.0
GNNML1	6.8 ± 3.7	15.2 ± 0.7	30.3 ± 1.2
GNNML3	61.5 ± 3.1	49.2 ± 3.6	36.3 ± 5.0
PPGN	-	-	-
Scattering (W1)	58.2 ± 0.8	45.6 ± 0.7	46.4 ± 1.3
Scattering (W2)	60.4 ± 0.5	49.5 ± 0.9	51.4 ± 1.0
BLIS-Net (W1)	63.1 ± 2.2	53.1 ± 1.3	54.8 ± 1.8
BLIS-Net (W2)	68.3 ± 2.1	56.3 ± 0.0	61.7 ± 2.6

Table 2: Accuracy on the PEMS03 traffic data set. PPGN results are not reported for the traffic dataset due to its prohibitively high computational and memory costs.

PEMS07	HOUR	DAY	WEEK
GCN	27.4 ± 2.0	14.6 ± 0.6	28.5 ± 0.5
GAT	33.2 ± 1.3	22.2 ± 1.2	36.5 ± 0.9
GIN	14.3 ± 12.6	15.8 ± 0.8	28.4 ± 0.6
GPS	39.9 ± 2.7	27.7 ± 1.9	30.4 ± 0.6
ChebNet	54.0 ± 4.2	61.4 ± 1.5	72.9 ± 3.3
GNNML1	8.0 ± 1.8	17.5 ± 1.7	28.3 ± 1.0
GNNML3	51.8 ± 5.3	52.3 ± 7.9	31.5 ± 4.1
PPGN	-	-	-
Scattering (W1)	54.0 ± 0.6	53.3 ± 0.9	56.9 ± 1.3
Scattering (W2)	54.3 ± 0.7	55.2 ± 1.2	61.6 ± 1.0
BLIS-Net (W1)	63.5 ± 1.1	72.9 ± 1.5	76.8 ± 2.0
BLIS-Net (W2)	63.4 ± 2.1	71.0 ± 2.4	77.3 ± 1.6

Table 3: Accuracy on the PEMS07 traffic data set.

from California's 3rd and 7th congressional districts and provide two months of consecutive traffic data collected between 2016 and 2018, depending on the data set[5]. We aim to predict the hour of the day (24 classes), the day of the week (7 classes), and the week of the month (4 classes) that each traffic observation corresponds to. On both PEMS03 (Table 2) and PEMS07 (Table 3), we observe that the message-passing-based methods (GCN, GAT, and GIN) perform poorly. Both wavelet-based methods (Scattering and BLIS-Net) perform well, with BLIS-Net outperforming Scattering perhaps because of its improved expressive power. The GPS graph transformer performs better than the message-passing based methods but less well than the wavelet-based methods, particularly when attempting to predict the week. ChebNet has the best performance while predicting the day and week for the PEMS03 dataset. However, for all other traffic datasets (including the two additional datasets in the appendix), BLIS-Net (and methods based off it) attain the top performance.

4.3 Partly Cloudy fMRI data set

Modeling functional magnetic resonance imaging (fMRI) data as a graph is a useful computational approach for analyzing brain signals; the nodes are brain regions of interest (ROIs) whose connections can be defined in multiple ways. The fMRI data provides graph signals in the form of a blood oxygenation level dependent (BOLD) signal across the ROIs.

Here, we utilize a data set collected from participants who were shown Disney Pixar's "Partly Cloudy" in Richardson et al. (2018). 39 ROIs were extracted from the fMRI data, and the graph connectivity was created using a k-nearest neighbors graph based on the centroids of the ROIs. We consider the problem of using the fMRI data to classify the emotional state of the animated film, delineating the frames into three classes, positive, negative, and neutral emotions. Similarly to Busch et al. (2023), we apply temporal smoothing at each node [6]. As seen in Table 4, BLIS-Net outperforms all other methods. As was generally observed in the PeMS data sets, scattering is the second best performing method followed by the GPS graph transformer, underscoring the value of capturing global information as well as the full frequency spectrum of the input signal.

[5] Additional experiments on data from the 4th and 8th district are provided in the supplement along with additional details on our experimental setup.

[6] Results without smoothing are in the supplement.

Partly Cloudy	Emotion classification
GCN	39.3 ± 5.9
GAT	40.6 ± 6.1
GIN	42.1 ± 6.0
GPS	56.4 ± 4.3
ChebNet	45.7 ± 5.9
GNNML1	48.8 ± 5.4
GNNML3	45.2 ± 5.6
PPGN	42.0 ± 5.3
Scattering (W1)	60.6 ± 4.9
Scattering (W2)	62.3 ± 5.1
BLIS-Net (W1)	$\mathbf{67.1 \pm 4.3}$
BLIS-Net (W2)	$\mathbf{68.3 \pm 3.6}$

Table 4: Accuracy on Partly Cloudy fMRI data.

5 CONCLUSION AND FUTURE WORK

We have introduced BLIS-Net, a network for processing graph signals. The key piece of our architecture is the BLIS module, which modifies the geometric scattering transform in several ways in order to provably increase its expressive power for signal classification. We then show that BLIS-Net achieves superior performance to both the original geometric scattering transform and other GNNs on both real and synthetic data.

There are also several natural avenues of future work. As alluded to in the introduction, Bodmann and Emilsdottir (2022) used a statistical analysis of the geometric scattering coefficients to detect anomalous traffic patterns. It is likely that similar tools can be used in conjunction with BLIS to detect anomalous signals on graphs such as traffic networks and brain-scan networks. Additionally, we note that many of our data sets have an implicit temporal structure in addition to the graph structure. Therefore, developing a space-time version of BLIS, perhaps using techniques inspired by Pan et al. (2021), would be a natural future direction. Lastly, we note that Wenkel et al. (2022) constructs a hybrid network which combines aspects of geometric scattering with more standard GCN-style networks and utilizes a localized attention mechanism to balance the two. We view hybridizations similar to this, with BLIS in place of the geometric scattering transform, as a potential avenue for improved numerical performance in future work.

References

Antonello, J. and Verhaegen, M. (2015). Modal-based phase retrieval for adaptive optics. *JOSA A*, 32(6):1160–1170.

Balan, R., Casazza, P., and Edidin, D. (2006). On signal reconstruction without phase. *Applied and Computational Harmonic Analysis*, 20(3):345–356.

Balcilar, M., Héroux, P., Gauzere, B., Vasseur, P., Adam, S., and Honeine, P. (2021). Breaking the limits of message passing graph neural networks. In *International Conference on Machine Learning*, pages 599–608. PMLR.

Bandeira, A. S., Cahill, J., Mixon, D. G., and Nelson, A. A. (2014). Saving phase: Injectivity and stability for phase retrieval. *Applied and Computational Harmonic Analysis*, 37(1):106–125.

Bhaskar, D., Grady, J. D., Castro, E., Perlmutter, M., and Krishnaswamy, S. (2022). Molecular graph generation via geometric scattering. In *32nd IEEE International Workshop on Machine Learning for Signal Processing, MLSP 2022, Xi'an, China, August 22-25, 2022*, pages 1–6. IEEE.

Bodmann, B. G. and Emilsdottir, I. (2022). A scattering transform for graphs based on heat semigroups, with an application for the detection of anomalies in positive time series with underlying periodicities. *arXiv:2208.12773*.

Bruna, J. and Mallat, S. (2019). Multiscale sparse microcanonical models. *Mathematical Statistics and Learning*, 1(3):257–315.

Busch, E. L., Huang, J., Benz, A., Wallenstein, T., Lajoie, G., Wolf, G., Krishnaswamy, S., and Turk-Browne, N. B. (2023). Multi-view manifold learning of human brain-state trajectories. *Nature Computational Science*, 3(3):240–253.

Cahill, J., Casazza, P., and Daubechies, I. (2016). Phase retrieval in infinite-dimensional hilbert spaces. *Transactions of the American Mathematical Society, Series B*, 3(3):63–76.

Castro, E., Benz, A., Tong, A., Wolf, G., and Krishnaswamy, S. (2020). Uncovering the Folding Landscape of RNA Secondary Structure Using Deep Graph Embeddings. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 4519–4528.

Chen, C., Petty, K., Skabardonis, A., Varaiya, P., and Jia, Z. (2001). Freeway performance measurement system: mining loop detector data. *Transportation Research Record*, 1748(1):96–102.

Cheng, C., Daubechies, I., Dym, N., and Lu, J. (2021). Stable phase retrieval from locally stable and conditionally connected measurements. *Applied and Computational Harmonic Analysis*, 55:440–465.

Chew, J., Hirn, M. J., Krishnaswamy, S., Needell, D., Perlmutter, M., Steach, H. R., Viswanath, S., and Wu, H. (2022). Geometric scattering on measure spaces. arXiv:2208.08561.

Chung, F. R. (1997). *Spectral graph theory*, volume 92. American Mathematical Soc.

Coifman, R. R. and Wickerhauser, M. V. (1992). Entropy-based algorithms for best basis selection. *IEEE Transactions on information theory*, 38(2):713–718.

Defferrard, M., Bresson, X., and Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems 29*, pages 3844–3852.

Errica, F., Podda, M., Bacciu, D., and Micheli, A. (2019). A fair comparison of graph neural networks for graph classification. *arXiv preprint arXiv:1912.09893*.

Fienup, C. and Dainty, J. (1987). Phase retrieval and image reconstruction for astronomy. *Image recovery: theory and application*, 231:275.

Gama, F., Bruna, J., and Ribeiro, A. (2019a). Stability of graph scattering transforms. In *Advances in Neural Information Processing Systems 33*.

Gama, F., Ribeiro, A., and Bruna, J. (2019b). Diffusion scattering transforms on graphs. In *International Conference on Learning Representations*.

Gao, F., Wolf, G., and Hirn, M. (2019). Geometric scattering for graph data analysis. In *Proceedings of the 36th International Conference on Machine Learning, PMLR*, volume 97, pages 2122–2131.

Guo, S., Lin, Y., Feng, N., Song, C., and Wan, H. (2019). Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33 No. 01, pages 922–929.

Hu, W., Liu, B., Gomes, J., Zitnik, M., Liang, P., Pande, V. S., and Leskovec, J. (2020). Strategies for pre-training graph neural networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Iwen, M. A., Merhi, S., and Perlmutter, M. (2019). Lower lipschitz bounds for phase retrieval from locally supported measurements. *Applied and Computational Harmonic Analysis*, 47(2):526–538.

Kipf, T. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *Proc. of ICLR*.

Li, X., Zhou, Y., Dvornek, N., Zhang, M., Gao, S., Zhuang, J., Scheinost, D., Staib, L. H., Ventola, P., and Duncan, J. S. (2021). BrainGNN: Interpretable brain graph neural network for fMRI analysis. *Medical Image Analysis*, 74:102233.

Liu, Z.-C., Xu, R., and Dong, Y.-H. (2012). Phase retrieval in protein crystallography. *Acta Crystallographica Section A: Foundations of Crystallography*, 68(2):256–265.

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

Mallat, S. and Waldspurger, I. (2015). Phase retrieval for the Cauchy wavelet transform. *Journal of Fourier Analysis and Applications*, 21(6):1251–1309.

Maron, H., Ben-Hamu, H., Serviansky, H., and Lipman, Y. (2019). Provably powerful graph networks. *Advances in neural information processing systems*, 32.

Ménoret, M., Farrugia, N., Pasdeloup, B., and Gripon, V. (2017). Evaluating graph signal processing for neuroimaging through classification and dimensionality reduction. In *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 618–622. IEEE.

Min, Y., Wenkel, F., Perlmutter, M., and Wolf, G. (2022). Can hybrid geometric scattering networks help solve the maximal clique problem? In *Neural Information Processing Symposium (NeurIPS)*.

Moon, K. R., van Dijk, D., Wang, Z., Gigante, S., Burkhardt, D., Chen, W. S., Yim, K., van den Elzen, A., Hirn, M. J., Coifman, R. R., Ivanova, N. B., Wolf, G., and Krishnaswamy, S. (2019). Visualizing structure and transitions for biological data exploration. *Nature BioTechnology*.

Pan, C., Chen, S., and Ortega, A. (2021). Spatiotemporal graph scattering transform. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

Perlmutter, M., He, J., Iwen, M., and Hirn, M. (2021). A hybrid scattering transform for signals with isolated singularities. In *2021 55th Asilomar Conference on Signals, Systems, and Computers*, pages 1322–1329. IEEE.

Perlmutter, M., Tong, A., Gao, F., Wolf, G., and Hirn, M. (2023). Understanding graph neural networks with generalized geometric scattering transforms. *SIAM Journal on Mathematics of Data Science (SIMODS), (to appear)*.

Rampášek, L., Galkin, M., Dwivedi, V. P., Luu, A. T., Wolf, G., and Beaini, D. (2022). Recipe for a general, powerful, scalable graph transformer. *Advances in*

Neural Information Processing Systems, 35:14501–14515.

Richardson, H., Lisandrelli, G., Riobueno-Naylor, A., and Saxe, R. (2018). Development of the social brain from age three to twelve years. *Nature communications*, 9(1):1027.

Rieck, B., Yates, T., Bock, C., Borgwardt, K., Wolf, G., Turk-Browne, N., and Krishnaswamy, S. (2020). Uncovering the topology of time-varying fmri data using cubical persistence. *Advances in neural information processing systems*, 33:6900–6912.

Shuman, D. I., Narang, S. K., Frossard, P., Ortega, A., and Vandergheynst, P. (2013). The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 30(3):83–98.

Tong, A., Wenkel, F., Bhaskar, D., Macdonald, K., Grady, J., Perlmutter, M., Krishnaswamy, S., and Wolf, G. (2022). Learnable filters for geometric scattering modules. *arXiv:2208.07458*.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. (2018). Graph attention networks. In *International Conference on Learning Representations*.

Wenkel, F., Min, Y., Hirn, M., Perlmutter, M., and Wolf, G. (2022). Overcoming oversmoothness in graph convolutional networks via hybrid scattering networks. *arXiv:2201.08932*.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Philip, S. Y. (2020). A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24.

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

Zhang, M. and Chen, Y. (2018). Link prediction based on graph neural networks. *Advances in neural information processing systems*, 31.

Zhu, J., Yan, Y., Zhao, L., Heimann, M., Akoglu, L., and Koutra, D. (2020). Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in neural information processing systems*, 33:7793–7804.

Zou, D. and Lerman, G. (2019a). Encoding robust representation for graph generation. In *International Joint Conference on Neural Networks*.

Zou, D. and Lerman, G. (2019b). Graph convolutional neural networks via scattering. *Applied and Computational Harmonic Analysis*, 49(3)(3):1046–1074.

Checklist

1. For all models and algorithms presented, check if you include:

   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [**Yes**/No/Not Applicable]

   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [**Yes**/No/Not Applicable]

   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [**Yes**/No/Not Applicable]

2. For any theoretical claim, check if you include:

   (a) Statements of the full set of assumptions of all theoretical results. [**Yes**/No/Not Applicable]

   (b) Complete proofs of all theoretical results. [**Yes**/No/Not Applicable]

   (c) Clear explanations of any assumptions. [**Yes**/No/Not Applicable]

3. For all figures and tables that present empirical results, check if you include:

   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [**Yes**/No/Not Applicable]

   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [**Yes**/No/Not Applicable]

   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [**Yes**/No/Not Applicable]

   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [**Yes**/No/Not Applicable]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

   (a) Citations of the creator If your work uses existing assets. [**Yes**/No/Not Applicable]

   (b) The license information of the assets, if applicable. [Yes/No/**Not Applicable**]

   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes/No/**Not Applicable**]

   (d) Information about consent from data providers/curators. [Yes/No/**Not Applicable**]

   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Yes/No/**Not Applicable**]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

   (a) The full text of instructions given to participants and screenshots. [Yes/No/**Not Applicable**]

   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Yes/No/**Not Applicable**]

   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Yes/No/**Not Applicable**]

Appendix

A Proof of Proposition 2.1

The proof of Proposition 2.1 follows by adapting the techniques used to prove Theorem 1 of Tong et al. (2022) and Proposition 2.2 of Perlmutter et al. (2023) to more general scales (for $\mathcal{W}^{(1)}$) and more general diffusion matrices (for $\mathcal{W}^{(2)}$).

Proof. We will first prove (6), which estabilished that $\mathcal{W}_J^{(1)}$ is an isometry. Note that by (2) and (4) we have

$$\Psi_j^{(1)} = q_j(K) = W^{-1}Vq_j(\Lambda)V^TW$$

for all $0 \leq j \leq J$ and also that

$$\Phi_J^{(1)} = q_{J+1}(K) = W^{-1}Vq_{J+1}(\Lambda)V^TW.$$

Additionally, we note that since V is unitary, the definition of $\langle\cdot,\cdot\rangle_{\mathbf{w}}$ implies that we have

$$\|\Psi_j^{(1)}\mathbf{x}\|_{\mathbf{w}}^2 = \langle\Psi_j^{(1)}\mathbf{x}, \Psi_j^{(1)}\mathbf{x}\rangle_{\mathbf{w}} = \mathbf{x}^TW^TVq_j(\Lambda)^2V^TW\mathbf{x},$$

and similarly $\|\Phi_j^{(1)}\mathbf{x}\|_{\mathbf{w}}^2 = \mathbf{x}^TW^TVq_{J+1}(\Lambda)^2V^TW\mathbf{x}$. Therefore

$$\begin{aligned}
\|\mathcal{W}_J^{(1)}\mathbf{x}\|_{\mathbf{w}}^2 &= \sum_{j=0}^{J}\|\Psi_j^{(1)}\mathbf{x}\|_{\mathbf{w}}^2 + \|\Phi_J^{(1)}\mathbf{x}\|_{\mathbf{w}}^2 \\
&= \mathbf{x}^TW^TV\left[\sum_{j=0}^{J+1}q_j(\Lambda)^2\right]V^TW\mathbf{x} \\
&= \mathbf{x}^TW^TVQ_J(\Lambda)V^TW\mathbf{x} \\
&= \langle Q_J(\Lambda)V^TW\mathbf{x}, V^TW\mathbf{x}\rangle_2,
\end{aligned}$$

where $Q_J(t) := \sum_{j=0}^{J+1} q_j(t)^2$ (and $Q_J(\Lambda)$ is defined term by term along the diagonal according to (2)). Therefore the lower frame bound on $\mathcal{W}_J^{(1)}$ is given by

$$\begin{aligned}
c_J^{(1)} &:= \inf_{\mathbf{x}\neq 0}\frac{\|\mathcal{W}_J^{(1)}\mathbf{x}\|_{\mathbf{w}}^2}{\|\mathbf{x}\|_{\mathbf{w}}^2} \\
&= \inf_{\mathbf{x}\neq 0}\frac{\langle Q_J(\Lambda)V^TW\mathbf{x}, V^TW\mathbf{x}\rangle_2}{\|W\mathbf{x}\|_2^2} \\
&= \inf_{\mathbf{x}\neq 0}\frac{\langle Q_J(\Lambda)V^TW\mathbf{x}, V^TW\mathbf{x}\rangle_2}{\|V^TW\mathbf{x}\|_2^2} \qquad \text{(B.1)} \\
&= \inf_{\mathbf{y}\neq 0}\frac{\langle Q_J(\Lambda)\mathbf{y}, \mathbf{y}\rangle}{\|\mathbf{y}\|_2^2} \qquad \text{(B.2)} \\
&= \min_{1\leq i\leq n} Q_J(\lambda_i). \qquad \text{(B.3)}
\end{aligned}$$

B.1 follows because V is unitary. B.2 follows because W is invertible and B.3 follows because $Q_J(\Lambda)$ is diagonal with nonzero entries $Q_J(\lambda_i)$. We can similarly calculate the upper frame bound to be

$$C_J^{(1)} := \sup_{\mathbf{x}\neq 0}\frac{\|\mathcal{W}_J^{(1)}\mathbf{x}\|_{\mathbf{w}}^2}{\|\mathbf{x}\|_{\mathbf{w}}^2} = \max_{1\leq i\leq n} Q_J(\lambda_i).$$

Using a telescoping sum, we see that

$$Q_J(t) = \sum_{j=0}^{J+1} q_j(t)^2 = \sum_{j=0}^{J+1} p_j(t) = t^{s_{J+1}} + \sum_{j=0}^{J} (t^{s_j} - t^{s_{j+1}}) = t^{s_{J+1}} - (t^{s_{J+1}} - 1) = 1 \tag{10}$$

uniformly on $0 \le t \le 1$. Thus, $c_J^{(1)} = C_J^{(1)} = 1$ which concludes the proof of (6).

Turning our attention to (7), we may use the same logic as before to see that the lower and upper frame bounds for $\mathcal{W}_J^{(2)}$ are given by:

$$c_J^{(2)} \coloneqq \inf_{\mathbf{x}\neq 0} \frac{\|\mathcal{W}_J^{(2)}\mathbf{x}\|_{\mathbf{w}}^2}{\|\mathbf{x}\|_{\mathbf{w}}^2} = \min_{1\le i\le n} P_J(\lambda_i), \quad C_J^{(2)} \coloneqq \sup_{\mathbf{x}\neq 0} \frac{\|\mathcal{W}_J^{(2)}\mathbf{x}\|_{\mathbf{w}}^2}{\|\mathbf{x}\|_{\mathbf{w}}^2} = \max_{1\le i\le n} P_J(\lambda_i),$$

where $P_J(t) \coloneqq \sum_{j=0}^{J+1} p_j(t)^2$. To determine the upper frame bound, it suffices to note that

$$\max_{1\le i\le n} P_J(\lambda_i) \le \sup_{t\in[0,1]} \sum_{j=0}^{J+1} p_j(t)^2 \le \sup_{t\in[0,1]} \left(\sum_{j=0}^{J+1} p_j(t)\right)^2 = 1$$

where the second inequality comes from the positivity of each $p_j(t)$ and the final equality comes from the same reasoning as in (10). For the lower bound, we see that

$$\min_{1\le i\le n} P_J(\lambda_i) \ge \inf_{t\in[0,1]} \sum_{j=0}^{J+1} p_j(t)^2 \ge \inf_{t\in[0,1]} p_0(t)^2 + p_{J+1}(t)^2 = \inf_{t\in[0,1]} (1-t)^2 + t^{2s_{J+1}}.$$

The final quantity is a positive constant depending only on s_{J+1}. Thus, this completes the proof.

□

Remark A.1. *In our experiments, we use dyadic scales, ($s_0 = 0, s_1 = 1, s_j = 2^{j-1}, j \ge 2$). In this case, Proposition 1 of Chew et al. (2022) implies that the lower frame bound c for the $\mathcal{W}_J^{(2)}$ wavelets may be chosen to be a universal constant.*

B Details and proof for Theorem 3.1

In this section, we provide full details on Theorem 3.1 as well as some discussion.

B.1 Background - Wavelet Phase Retrieval

The original scattering transform (Mallat, 2012) was introduced as a theoretical model for understanding the success of convolutional neural networks, defining scattering coefficients via an alternating sequence of wavelet convolutions and pointwise absolute values (moduluses):

$$S_J[j_1, \ldots, j_m]f = \Phi_J H\Psi_{j_m} \ldots H\Psi_{j_1} f, \quad \text{for } f \in \mathbf{L}^2(\mathbb{R}^n).$$

A natural question is to what extent do these coefficients determine a signal f? If two signals, f_1 and f_2 have the same coefficents, does this imply f_1 and f_2 coincide?

To answer this question, Mallat and Waldspurger (2015), studied the descriptive power of the wavelet-modulus, $M\Psi_j$ which is the key building block of the scattering transform. Since Ψ_j is linear, it is immediate that $H\Psi_j f_1 = H\Psi_j f_2$ whenever $f_1 = \pm f_2$ (or more generally when $f_1 = e^{i\theta} f_2$ in the case the functions are complex-valued). The question then becomes whether this is the only setting in which $H\Psi_j f_1 = H\Psi_j f_2$, i.e., are there any non-trivial ambiguities in the wavelet modulus. Questions such as this, whether or not a function can be determined (up to a global sign) by magnitude-only measurements, are known as the phase retrieval problems (Bandeira et al., 2014) and arise in wide variety of scientific domains including optics (Antonello and Verhaegen, 2015), astronomy Fienup and Dainty (1987), x-ray crystallography (Liu et al., 2012), and speech-signal processing (Balan et al., 2006).

The primary results of Mallat and Waldspurger (2015) are (i) if the Ψ_j are chosen to be Cauchy wavelets, then the wavelet modulus *is* injective (up to the equivalence relation $f(x) \sim e^{i\theta} f(x)$) and therefore invertible. (ii) There is no uniform modulus of continuity on the inverse map, i.e., there are signals f_1, f_2 which are far apart (in the quotient metric induced by the relevant equivalence relation) such that f_1 and f_2 have nearly identical wavelet modulus.

Theorem B.1 stated below, which is a more detailed version of Theorem 3.1 from the main body, is meant to address the analogous question in the graph setting. Is the wavelet modulus invertible (up to a global sign change)? We show that under certain circumstances the answer to this question is no. There are signals $\mathbf{x_1} \neq \pm\mathbf{x_2}$ with identical wavelet moduluses therefore indentical scattering coefficients.

Additionally, we also note that Theorem 3.2 is also partially motivated by the second result of Mallat and Waldspurger (2015). It shows that unlike the (Euclidean) scattering transform, the map which recovers a signal $\mathbf{x}$ from its BLIS coefficients is Lipschitz continuous. Therefore, the BLIS module can be stably inverted.

B.2 Statement and Proof of Theorem B.1

We first introduce some notation. For $v_i, v_j \in V$, we let $d(v_i, v_j)$ denote the unweighted path distance between v_i and v_j. That is, $d(v_i, v_j)$ is the smallest k such that $A^k(v_i, v_j) \neq 0$, when $v_i \neq v_j$, and $d(v_i, v_i) = 0$. We then define the diameter of G by

$$\mathrm{diam}(G) = \max_{v_i, v_j \in V} d(v_i, v_j).$$

With this notation, we may now state our theorem in detail and provide a proof.

Theorem B.1. *Let $\mathcal{W}_J = \{\Psi_j\}_{j=0}^J \cup \{\Phi_J\}$ be the wavelets $\mathcal{W}_J^{(2)}$ constructed in Section 2.4. Suppose at least one of the following two conditions hold.*

1. *G is a bipartite graph.*
2. *$g(t)$ is as in (3) and $\mathrm{diam}(G) > 2s_{J+1}$.*

Then there exist signals $\mathbf{x_1}, \mathbf{x_2}$ such that $\mathbf{x_1} \neq \pm\mathbf{x_2}$, but

$$H\Psi_j \mathbf{x_1} = H\Psi_j \mathbf{x_2} \quad \textit{for all } 0 \leq j \leq J,$$

and therefore, $\mathbf{x_1}$ and $\mathbf{x_2}$ have identical m-th order scattering coefficients for all $m \geq 1$.

Proof. Let us first consider the case where G is bipartite. As in Section 2.1, let $\mathbf{v_1}, \ldots, \mathbf{v_n}$ denote the eigenvalues of L_N with $L_N \mathbf{v_i} = \omega_i \mathbf{v_i}$, $0 = \omega_1 < \omega_2 \leq \ldots \leq \omega_n \leq 2$. It is known (see, e.g., Lemma 1.7 of Chung (1997)) that since G is bipartite, we have $\omega_n = 2$.

Since the function $g(t)$ defined in Section 2.3 satisfies $g(0) = 1$, $g(2) = 0$, this implies that $T = g(L_N)$ has eigenvalues of 1 and 0. Moreover, since K is similar to T, this implies that K also has eigenvalues 0 and 1. That is, there exist vectors $\mathbf{u_1}, \mathbf{u_2} \neq 0$ such that $K\mathbf{u_1} = \mathbf{u_1}$, $K\mathbf{u_2} = 0$.

Let $\mathbf{x_1} = \mathbf{u_1} + \mathbf{u_2}$ and $\mathbf{x_2} = \mathbf{u_1} - \mathbf{u_2}$. By definition, neither $\mathbf{u_1}$ or $\mathbf{u_2}$ are the zero vector and therefore it is clear that $\mathbf{x_1} \neq \pm\mathbf{x_2}$. Thus, the proof will be complete once we show that $H\Psi_j \mathrm{x}_1 = H\Psi_j \mathrm{x}_2$ for all $0 \leq j \leq J$.

We first note that for $i = 1, 2$ and $1 \leq j \leq J$, we have $s_i, s_{i+1} > 0$ and thus we have

$$\begin{aligned}
\Psi_j \mathbf{x_i} &= K^{s_j} \mathbf{x_i} - K^{s_{j+1}} \mathbf{x_i} \\
&= K^{s_j}(\mathbf{u_1} \pm \mathbf{u_2}) - K^{s_{j+1}}(\mathbf{u_1} \pm \mathbf{u_2}) \\
&= (K^{s_j} \mathbf{u_1} - K^{s_{j+1}} \mathbf{u_1}) \pm (K^{s_{j+1}} \mathbf{u_2} - K^{s_j} \mathbf{u_2}) \\
&= \mathbf{u_1} - \mathbf{u_1} \pm (0 - 0) \\
&= 0,
\end{aligned}$$

which implies that $H\Psi_j \mathbf{x_1} = H\Psi_j \mathbf{x_2} = 0$.

In the case where $j = 0$, we have $\Psi_0 = I - K$. Therefore,

$$\begin{aligned}\Psi_0\mathbf{x_i} &= \mathbf{x_i} - K\mathbf{x_i}\\ &= \mathbf{u_1} \pm \mathbf{u_2} - K(\mathbf{u_1} \pm \mathbf{u_2})\\ &= \mathbf{u_1} \pm \mathbf{u_2} - (\mathbf{u_1} \pm 0)\\ &= \pm\mathbf{u_2}.\end{aligned}$$

Therefore, we also have $H\Psi_0\mathbf{x_1} = H\Psi_0\mathbf{x_2}$, which completes the proof under the assumption the graph is bipartite.

In the case where $\text{diam}(G) > 2s_{J+1}$ and $g(t)$ is as in (3), the proof is based on adapting the techniques from Iwen et al. (2019) which analyzed the instability of phase retrieval from locally supported measurements on $\mathbb{C}^n$ to the irregular geometry of a graph. (See also Cahill et al. (2016) and Cheng et al. (2021).) The assumption that $\text{diam}(G) > 2s_{J+1}$ implies that there exist disjoint, non-empty subsets $S_1, S_2 \subseteq V$ such that

$$\min_{v_1\in S_2,\, v_2\in S_2} d(v_1, v_2) \geq 2s_{J+1} + 1. \tag{11}$$

We now define $\mathbf{x_1}$ and $\mathbf{x_2}$ by

$$\mathbf{x_1} \coloneqq \boldsymbol{\delta}_{S_1} \pm \boldsymbol{\delta}_{S_2} \quad \text{and} \quad \mathbf{x_2} \coloneqq \boldsymbol{\delta}_{S_1} \pm \boldsymbol{\delta}_{S_2}, \tag{12}$$

where $\boldsymbol{\delta}_{S_1}$ is the indicator signal defined by $\boldsymbol{\delta}_{S_1}(v) = 1$ if $v \in S_1$, $\boldsymbol{\delta}_{S_1}(v) = 0$ otherwise, and $\boldsymbol{\delta}_{S_2}$ is defined similarly.

The following lemma will imply that there is no overlap in the support of $\Psi_j\boldsymbol{\delta}_{S_1}$ with $\Psi_j\boldsymbol{\delta}_{S_2}$.

Lemma B.1. *Let $\mathbf{x}$ be a graph signal whose non-zero entries are contained a set $S \subseteq V$. Then for all integers $t \geq 0$, the support of $K^t\mathbf{x}$ is contained in the set $S_t = \{v \in V : \exists u \in S, d(u, v) \leq t\}$.*

Proof. We first show that all $K_{i,j}$ are zero except for when either $i = j$ or $\{i, j\} \in E$. Indeed, this property is clearly satisfied by the unnormalized Laplacian $L_U = D - A$. Morever, the non-zero entries of a matrix are unchanged by multiplication (either on the left of the right) by a diagonal matrix. Therefore, this property is also satisfied by $L_N = D^{-1/2}L_UD^{-1/2}$. Additionally, this property is clearly satisfied by the identity matrix and also preserved under linear combinations. Therefore, it is also satisfied by $T = g(L_N)$. Lastly, we note that is also satisfied by K since $K = W^{-1}TW$ and W is diagonal.

We now prove the lemma. The case where $t = 0$ is trivial. For $t = 1$, we note that $\{K\mathbf{x}_i\} \neq 0$ implies that there exist j such that $K_{i,j} \neq 0$ and $x_j \neq 0$. In light of the preceeding paragraph, this implies that v_i is within distance one of some point in S which proves the $t = 1$ case. The result now follows inductively noting that $K^{t+1}\mathbf{x} = K(K^t\mathbf{x})$ □

In light of Lemma B.1, we see that for all $0 \leq j \leq J$, the support of $\boldsymbol{\delta}_{S_1}$ is contained in $S_{1,s_{J+1}} \coloneqq \{v \in V : \exists u \in S_1, d(u, v) \leq s_{J+1}\}$ and the support of $\boldsymbol{\delta}_{S_2}$ is contained in $S_{2,s_{J+1}} \coloneqq \{v \in V : \exists u \in S_2, d(u, v) \leq s_{J+1}\}$. Therefore, since

$$\Psi_j(\mathbf{x_i}) = \Psi_j\boldsymbol{\delta}_{S_1} \pm \Psi_j\boldsymbol{\delta}_{S_2},$$

(11) implies that

$$\Psi_j(\mathbf{x_i})(v) = \begin{cases}\Psi_j(\boldsymbol{\delta}_{S_1})(v) & \text{if } v \in S_1\\ \pm\Psi_j(\boldsymbol{\delta}_{S_2})(v) & \text{if } v \in S_2\\ 0 & \text{otherwise}\end{cases}.$$

This implies that $H\Psi_j\mathbf{x_1}(v) = H\Psi_j\mathbf{x_2}(v)$ for all j and all v and therfore completes the proof.

□

Remark B.1. *In addition to $H\Psi_j\mathbf{x_1} = H\Psi_j\mathbf{x_2}$, we aslo have $H\Phi_J\mathbf{x_1} = H\Phi_J\mathbf{x_2}$. In the bipartite case, we have $\Phi_J\mathbf{x_1} = \Phi_J\mathbf{x_2} = \mathbf{u_1}$. Moreover, in the case where the graph has a large diameter, the fact that $H\Phi_J\mathbf{x_1} = H\Phi_J\mathbf{x_2}$ follows directly from Lemma B.1.*

Remark B.2. *In the large diameter case with $K = P$, one may also modify the above construction to ensure that the zeroth-order coefficients $\Phi_J \mathbf{x_i}$ are identical for $\mathbf{x_1}$ and $\mathbf{x_2}$, after they are fed into the aggregation module. To do this, we one chooses three sets S_1, S_2, S_3, which are all sufficiently far apart and satisfy $|S_1| = |S_2| = |S_3|$ and similar to the proof above set $\mathbf{x_1} = \boldsymbol{\delta}_{S_1} + \boldsymbol{\delta}_{S_2} - \boldsymbol{\delta}_{S_3}$, $\mathbf{x_2} = \boldsymbol{\delta}_{S_1} - \boldsymbol{\delta}_{S_2} + \boldsymbol{\delta}_{S_3}$. The fact that P is a Markov matrix implies that if preserves the ℓ^1 norm of each of the Dirac $\boldsymbol{\delta}$ functions. Therefore, one may verify that the aggregated zero-th order coefficients will be equal to $|S_1|$ for both signals. Additionally, we note that several papers on the graph scattering transform including Gama et al. (2019b) and Gao et al. (2019) use global summation rather than low-pass filtering in the definition of the scattering coefficients, i.e., $\overline{S}_J[j_1, \ldots, j_m]\mathbf{x} = \|U[j_1, \ldots, j_m]\mathbf{x}\|_1$. This case, the zero's order coefficients of these two signals will also coincide, regardless of the choice of diffusion matrix.*

Remark B.3. *Theorem B.1 provides two examples of settings where the wavelet modulus fails to be injective. We note that the first assumption, that the graph is bipartite, is satisfied by graphs used in recommender systems where there are links between users and products. The latter assumption, that the graph has a large diameter will typically be satisfied by graphs constructed from geo-spatial data which do not experience the small world phenomenon. Our analysis shows that if graphs with large diameter, one needs to choose a large value of J therefore increasing the computational cost of the wavelet transform. By contrast Theorem 3.2 shows that in the BLIS module, J may be chosen independent of the graph allowing for efficient implementation on graphs with large diameters.*

C Proof of Theorem 3.2

Theorem 3.2 is proved by iteratively applying the frame bounds (5) as well as the following lemma which shows that the map $\sigma : \mathbb{R}^n \to \mathbb{R}^{2n}$ defined by

$$\sigma(\mathbf{x}) = (\sigma_1(\mathbf{x})^T, \sigma_2(\mathbf{x})^T)^T$$

is bi-Lipschitz.

Lemma C.1. *For all $\mathbf{x}$ and $\mathbf{y}$ in $\mathbb{R}^n$, we have*

$$\frac{1}{2}\left\|\mathbf{x}-\mathbf{y}\right\|_{\mathbf{w}}^2 \leq \left\|\sigma_1(\mathbf{x}) - \sigma_1(\mathbf{y})\right\|_{\mathbf{w}}^2 + \left\|\sigma_2(\mathbf{x}) - \sigma_2(\mathbf{y})\right\|_{\mathbf{w}}^2 \leq \left\|\mathbf{x}-\mathbf{y}\right\|_{\mathbf{w}}^2.$$

For a proof of Lemma C.1, please see Appendix F.

Proof of Theorem 3.2. We argue by induction on m. To establish the base case $m = 1$, we note that

$$\left\|\mathbf{B}_1(\mathbf{x}) - \mathbf{B}_1(\mathbf{y})\right\|_{\mathbf{w},2}^2 = \sum_{k=1}^{2}\sum_{j=0}^{J+1} \left\|B[j,k](\mathbf{x}) - B[j,k](\mathbf{y})\right\|_{\mathbf{w}}^2 = \sum_{k=1}^{2}\sum_{j=0}^{J+1} \left\|\sigma_k(F_j\mathbf{x}) - \sigma_k(F_j\mathbf{y})\right\|_{\mathbf{w}}^2.$$

Lemma C.1 and (5) imply

$$\begin{aligned}
\frac{c}{2}\|\mathbf{x}-\mathbf{y}\|_{\mathbf{w}} \leq & \frac{1}{2}\sum_{j=0}^{J+1} \left\|F_j\mathbf{x} - F_j\mathbf{y}\right\|_{\mathbf{w}}^2 \\
& \leq \sum_{j=0}^{J+1}\sum_{k=1}^{2} \left\|\sigma_k(F_j\mathbf{x}) - \sigma_k(F_j\mathbf{y})\right\|_{\mathbf{w}}^2 \\
& \leq \sum_{j=0}^{J+1} \left\|F_j\mathbf{x} - F_j\mathbf{y}\right\|_{\mathbf{w}}^2 \\
& \leq C\|\mathbf{x}-\mathbf{y}\|_{\mathbf{w}},
\end{aligned}$$

which establishes the base case.

Now, assume the result for m, i.e.,

$$\left(\frac{c}{2}\right)^m \left\|\mathbf{x}-\mathbf{y}\right\|_{\mathbf{w}}^2 \leq \left\|\mathbf{B}_m(\mathbf{x}) - \mathbf{B}_m(\mathbf{y})\right\|_{\mathbf{w},2}^2 \leq C^m \left\|\mathbf{x}-\mathbf{y}\right\|_{\mathbf{w}}^2,$$

and consider $\|\mathbf{B}_{m+1}(\mathbf{x})-\mathbf{B}_{m+1}(\mathbf{y})\|^2_{\mathbf{w},2}$. We note that by construction we have

$$B[j_1,k_1,\ldots,j_m,k_m,j_{m+1},k_{m+1}](\mathbf{x})=B[j_{m+1},k_{m+1}]B[j_1,k_1,\ldots,j_m,k_m](\mathbf{x}).$$

Therefore, for any fixed $j_1,k_1,\ldots,j_m,k_m$, we have

$$\begin{aligned}
&\sum_{k_{m+1}=1}^{2}\sum_{j_{m+1}=0}^{J}\|B[j_1,k_1,\ldots,j_m,k_m,j_{m+1},k_{m+1}](\mathbf{x})-B[j_1,k_1,\ldots,j_m,k_m,j_{m+1},k_{m+1}](\mathbf{y})\|^2_{\mathbf{w}}\\
=&\sum_{k_{m+1}=1}^{2}\sum_{j_{m+1}=0}^{J}\|B[j_{m+1},k_{m+1}]B[j_1,k_1,\ldots,j_m,k_m](\mathbf{x})-B[j_{m+1},k_{m+1}]B[j_1,k_1,\ldots,j_m,k_m](\mathbf{y})\|^2_{\mathbf{w}}\\
\leq& C\|B[j_1,k_1,\ldots,j_m,k_m](\mathbf{x})-B[j_1,k_1,\ldots,j_m,k_m](\mathbf{y})\|^2_{\mathbf{w}},
\end{aligned}\tag{13}$$

where the final inequality follows by applying the result with $m=1$. Similarly we have

$$\begin{aligned}
&\sum_{k_{m+1}=1}^{2}\sum_{j_{m+1}=0}^{J}\|B[j_1,k_1,\ldots,j_m,k_m,j_{m+1},k_{m+1}](\mathbf{x})-B[j_1,k_1,\ldots,j_m,k_m,j_{m+1},k_{m+1}](\mathbf{y})\|^2_{\mathbf{w}}\\
\geq&\frac{c}{2}\|B[j_1,k_1,\ldots,j_m,k_m](\mathbf{x})-B[j_1,k_1,\ldots,j_m,k_m](\mathbf{y})\|^2_{\mathbf{w}}.
\end{aligned}\tag{14}$$

Therefore, using the inductive hypothesis and (13), we have

$$\begin{aligned}
&\|\mathbf{B}_{m+1}(\mathbf{x})-\mathbf{B}_{m+1}(\mathbf{y})\|^2_{\mathbf{w},2}\\
=&\sum_{k_{m+1}=1}^{2}\sum_{j_{m+1}=0}^{J+1}\sum_{k_m=1}^{2}\sum_{j_m=0}^{J+1}\cdots\sum_{k_1=1}^{2}\sum_{j_1=0}^{J+1}\|B[j_1,k_1,\ldots,j_m,k_m,j_{m+1},k_{m+1}](\mathbf{x})-B[j_1,k_1,\ldots,j_m,k_m,j_{m+1},k_{m+1}](\mathbf{y})\|^2_{\mathbf{w}}\\
\leq& C\sum_{k_m=1}^{2}\sum_{j_m=0}^{J+1}\cdots\sum_{k_1=1}^{2}\sum_{j_1=0}^{J+1}\|B[j_1,k_1,\ldots,j_m,k_m,j_{m+1},k_{m+1}](\mathbf{x})-B[j_1,k_1,\ldots,j_m,k_m,j_{m+1},k_{m+1}](\mathbf{y})\|^2_{\mathbf{w}}\\
\leq& C^{m+1}\|\mathbf{x}-\mathbf{y}\|^2_{\mathbf{w}}.
\end{aligned}$$

which completes the proof for the upper bound. The lower bound follows by the same reasoning, but with (14) in place of (13).

□

C.1 Inverting the BLIS Module

The lower bound in Theorem 3.2 implies the existence of a Lipschitz continuous inverse map (defined on the range of $\mathbf{B}_m$) which recovers $\mathbf{x}$ from $\mathbf{B}_m(\mathbf{x})$. This is noteworthy in part because because numerous works such as Zou and Lerman (2019a); Bhaskar et al. (2022); Perlmutter et al. (2021); Bruna and Mallat (2019) have attempted to invert variations of the scattering transform for the purposes of data synthesis (with varying degrees of theoretical justification). We also note that in the the case where the wavelets are chosen to be $\mathcal{W}_J^{(2)}$ it is straightforward to invert each layer of the BLIS module since $\mathbf{x}=\sigma_1(\mathbf{x})-\sigma_2(\mathbf{x})$ and $\mathbf{x}=\sum_{j=0}^{J}\Psi_j^{(2)}\mathbf{x}+\Phi_J^{(2)}\mathbf{x}$. Indeed, in this setting, each layer of the BLIS module can essentially be viewed as a decomposition of the input signal somewhat analogous to wavelet packets (see e.g., Coifman and Wickerhauser (1992)) .

D Proof of Theorem 3.3

Proof. Let Π be a permutation matrix and let A', D', L'_N, W', K', etc, be the analogs of A, D, L_N, W, and K after the permutation.

One may verify that $A'=\Pi A\Pi^T$, and $D'=\Pi A\Pi^T$ (where one Π permutes the rows and the other permutes the columns). Since $\Pi^T\Pi=I=\Pi\Pi^T$, we see that $(D')^{1/2}=\Pi D^{1/2}\Pi^T$. Therefore,

$$L'_N=I-(\Pi D^{-1/2}\Pi^T)(\Pi A\Pi^T)(\Pi D^{-1/2}\Pi^T)=\Pi L_N\Pi^T.$$

Thus, we may compute

$$L_N'(\Pi\mathbf{v_i}) = \Pi L_N \Pi^T(\Pi\mathbf{v_i}) = \lambda_i \Pi\mathbf{v_i},$$

which implies that $\lambda_i' = \lambda_i$, $\mathbf{v_i}' = \Pi\mathbf{v_i}$, and therefore that the eigendecomposition of L_N' is given by

$$L_N' = (\Pi V)\Omega(\Pi V)^T.$$

Thus, we have

$$T' = g(L_N') = (\Pi V)g(\Omega)(\Pi V)^T = (\Pi V)g(\Omega)V^T\Pi^T = \Pi T \Pi^T.$$

Additionally, for a suitably nice function h (chosen to be either p_j or q_j), we have

$$h(T') = \Pi V h(g(\Omega))V^T\Pi^T = \Pi h(T)\Pi^T.$$

In particular, for either family of wavelets ($\mathcal{W}_J^{(1)}$ or $\mathcal{W}_J^{(2)}$), and we have $F_j' = \Pi F_j \Pi^T$, where, as in Section 3.1, F_j is a generic member of the frame. Additionally, both σ_1 and σ_2 are element wise operators and thus commute with permutations. Therefore, we have

$$\begin{aligned} B'[j_1,k_1,\ldots,j_m,k_m] &= \sigma_m(\Pi F_{j_m}\Pi^T(\sigma_{m-1}\ldots\sigma_1(\Pi F_{j_1}\Pi^T\cdot)\ldots) \\ &= \Pi B[j_1,k_1,\ldots,j_m,k_m]\Pi^T. \end{aligned}$$

This leads us to

$$B'[j_1,k_1,\ldots,j_m,k_m]\Pi\mathbf{x} = \Pi B[j_1,k_1,\ldots,j_m,k_m]\mathbf{x}$$

as desired.

□

E Proof of Theorem 3.4

The proof of Theorem 3.4 is nearly identical to that of Theorem 3.2, but relies on the following lemma in place of Lemma C.1.

Lemma E.1. *For all* $\mathbf{x} \in \mathbb{R}^n$*, we have*

$$\|\sigma_1(\mathbf{x})\|_{\mathbf{w}}^2 + \|\sigma_2(\mathbf{x})\|_{\mathbf{w}}^2 = \|\mathbf{x}\|_{\mathbf{w}}^2.$$

For a proof of Lemma E.1, please see Appendix G.

Proof of Theorem 3.4. We proceed inductively on m. In the case $m = 1$, we apply Lemma E.1 to see

$$\|\mathbf{B}_1(\mathbf{x})\|_{\mathbf{w},2}^2 = \sum_{k=1}^{2}\sum_{j=0}^{J+1}\|B[j,k](\mathbf{x})\|_{\mathbf{w}}^2 = \sum_{j=0}^{J+1}\sum_{k=1}^{2}\|\sigma_k(F_j\mathbf{x})\|_{\mathbf{w}}^2 = \sum_{j=0}^{J+1}\|F_j\mathbf{x}\|_{\mathbf{w}}^2$$

Therefore, by (5) we have

$$c\|\mathbf{x}\|_{\mathbf{w}}^2 \le \|\mathbf{B}_1(\mathbf{x})\|_{\mathbf{w},2}^2 \le C\|\mathbf{x}\|_{\mathbf{w}}^2,$$

which establishes the claim in the base case $m = 1$.

Now, assume the result for m, i.e., $c^m\|\mathbf{x}\|_{\mathbf{w}}^2 \le \|\mathbf{B}_m(\mathbf{x})\|_{\mathbf{w},2}^2 \le C^m\|\mathbf{x}\|_{\mathbf{w}}^2$, and consider $\|\mathbf{B}_{m+1}(\mathbf{x})\|_{\mathbf{w},2}^2$. As in the proof of Theorem 3.2, we have

$$B[j_1,k_1,\ldots,j_m,k_m,j_{m+1},k_{m+1}](\mathbf{x}) = B[j_{m+1},k_{m+1}]B[j_1,k_1,\ldots,j_m,k_m](\mathbf{x}).$$

Therefore, for any fixed $j_1,k_1\ldots,j_m,k_m$, we have

$$\begin{aligned} \sum_{k_{m+1}=1}^{2}\sum_{j_{m+1}=0}^{J}\|B[j_1,k_1,\ldots,j_m,k_m,j_{m+1},k_{m+1}](\mathbf{x})\|_{\mathbf{w}}^2 &= \sum_{k_{m+1}=1}^{2}\sum_{j_{m+1}=0}^{J}\|B[j_{m+1},k_{m+1}]B[j_1,k_1,\ldots,j_m,k_m](\mathbf{x})\|_{\mathbf{w}}^2 \\ &\le C\|B[j_1,k_1,\ldots,j_m,k_m](\mathbf{x})\|_{\mathbf{w}}^2, \end{aligned}$$

where the final inequality follows by applying the result with $m = 1$. Similarly,

$$\sum_{k_{m+1}=1}^{2} \sum_{j_{m+1}=0}^{J} \|B[j_1, k_1, \ldots, j_m, k_m, j_{m+1}, k_{m+1}](\mathbf{x})\|_{\mathbf{w}}^2 \geq c\|B[j_1, k_1, \ldots, j_m, k_m](\mathbf{x})\|_{\mathbf{w}}^2.$$

Therefore, using the inductive hypothesis, we have

$$\begin{aligned}\|\mathbf{B}_{m+1}(\mathbf{x})\|_{\mathbf{w},2}^2 = &\sum_{k_{m+1}=1}^{2} \sum_{j_{m+1}=0}^{J+1} \sum_{k_m=1}^{2} \sum_{j_m=0}^{J+1} \cdots \sum_{k_1=1}^{2} \sum_{j_1=0}^{J+1} \|B[j_1, k_1, \ldots, j_m, k_m, j_{m+1}, k_{m+1}](\mathbf{x})\|_{\mathbf{w}}^2 \\ \leq &C \sum_{k_m=1}^{2} \sum_{j_m=0}^{J+1} \cdots \sum_{k_1=1}^{2} \sum_{j_1=0}^{J+1} \|B[j_1, k_1, \ldots, j_m, k_m, j_{m+1}, k_{m+1}](\mathbf{x})\|_{\mathbf{w}}^2 \\ \leq &C^{m+1}\|\mathbf{x}\|_{\mathbf{w}}^2,\end{aligned}$$

and the lower bound follows similarly.

□

F Proof of Lemma C.1

Proof. It suffices to show that for all $a, b \in \mathbb{R}$ we have

$$\frac{1}{2}|a-b|^2 \leq |\sigma_1(a) - \sigma_1(b)|^2 + |\sigma_2(a) - \sigma_2(b)|^2 \leq |a-b|^2. \tag{15}$$

For then we will have,

$$\begin{aligned}&\frac{1}{2}\|\mathbf{x}-\mathbf{y}\|_{\mathbf{w}}^2 \\ =&\frac{1}{2}\sum_{i=1}^{n} |x_i - y_i|^2 w_i \\ \leq&\sum_{i=1}^{n} |\sigma_1(x_i) - \sigma_1(y_i)|^2 w_i + \sum_{i=1}^{n} |\sigma_2(x_i) - \sigma_2(y_i)|^2 w_i \qquad (16) \\ \leq&\sum_{i=1}^{n} |x_i - y_i|^2 w_i \\ =&\|\mathbf{x}-\mathbf{y}\|_{\mathbf{w}},\end{aligned}$$

which will complete the proof since the term from (16) is exactly $\|\sigma_1(\mathbf{x}-\mathbf{y})\|_{\mathbf{w}}^2 + \|\sigma_1(\mathbf{x}-\mathbf{y})\|_{\mathbf{w}}^2$.

To prove (15), we note that in the case where a and b have the same sign, then either $\sigma_1(a) = |a|$, $\sigma_1(b) = |b|$, and $\sigma_2(a) = \sigma_2(b) = 0$ or $\sigma_2(a) = |a|$, $\sigma_2(b) = |b|$, and $\sigma_1(a) = \sigma_1(b) = 0$. Either way, we have

$$|\sigma_1(a) - \sigma_1(b)|^2 + |\sigma_2(a) - \sigma_2(b)|^2 = |a-b|^2.$$

In the case where a and b have different signs, assume without loss of generality that $a \geq 0 \geq b$. Then, $|a-b| = |a| + |b|$ and so the result follows from noting

$$|\sigma_1(a) - \sigma_1(b)|^2 + |\sigma_2(a) - \sigma_2(b)|^2 = |a|^2 + |b|^2 \geq \frac{1}{2}(|a| + |b|)^2$$

as well as the fact that $|a|^2 + |b|^2 \leq (|a| + |b|)^2$.

□

G Proof of Lemma E.1

Proof. Let $\mathbf{x} \in \mathbb{R}^n$. Let $\mathcal{I} \coloneqq \{i : x_i \neq 0\}$ and note that we may write $\mathcal{I}$ as the disjoint union $\mathcal{I} = \mathcal{I}_1 \cup \mathcal{I}_2$ where $\mathcal{I}_1 \coloneqq \{i : (\sigma_1(\mathbf{x}))_i \neq 0\}$, $\mathcal{I}_2 \coloneqq \{i : (\sigma_1(\mathbf{x}))_i \neq 0\}$. Observe that for $j = 1, 2$ we have $|(\sigma_j(\mathbf{x}))_i|^2 = |x_i|^2$ whenever $i \in \mathcal{I}_j$. Therefore,

$$\|\mathbf{x}\|_{\mathbf{w}}^2 = \sum_{i \in \mathcal{I}_1} |x_i|^2 w_i + \sum_{i \in \mathcal{I}_2} |x_i|^2 w_i = \sum_{i \in \mathcal{I}_1} |(\sigma_1(\mathbf{x}))_i|^2 w_i + \sum_{i \in \mathcal{I}_2} |(\sigma_2(\mathbf{x}))_i|^2 w_i = \|\sigma_1(\mathbf{x})\|_{\mathbf{w}}^2 + \|\sigma_2(\mathbf{x})\|_{\mathbf{w}}^2. \qquad \square$$

H BLIS-Net Implementation and Computational Complexity

Here we will discuss the implementation used in our experiments and also a modified implementation that can be used to increase the scalability of our network to large graphs with sparse connectivity.

In our experiments, we chose the diffusion operator $K = P = \frac{1}{2}(I + AD^{-1})$ with dyadic scales. When using $\mathcal{W}^{(2)}$, we computed the powers P^{2^j} iteratively using the formula $P^{2^{j+1}} = P^{2^j} P^{2^j}$ and then computed the wavelets via subtraction. For the $\mathcal{W}_J^{(1)}$ wavelets, we computed an eigendecomposition of T and then applied the functions q_j along the diagonal. This simple implementation requires $\mathcal{O}(Jn^3)$ flops to construct the wavelet matrices for $\mathcal{W}_J^{(2)}$ and $\mathcal{O}(n^3 + Jn)$ for $\mathcal{W}_J^{(1)}$. Then to perform the wavelet transform via matrix-vector multiplication we incur a cost of $\mathcal{O}(Jn^2)$ for each signal. Thus if there are N signals in the data set, the total cost of the wavelet transform is $\mathcal{O}(Jn^3 + Jn^2N)$. The memory cost of storing the wavelet matrices is $\mathcal{O}(Jn^2)$ for $\mathcal{W}_J^{(2)}$ and $\mathcal{O}(n^2 + Jn)$ for $\mathcal{W}_J^{(1)}$. Since BLIS module consists of m iterations of the wavelets followed by σ_1 and σ_2, it follows that the computational cost of an m-layer BLIS module is $\mathcal{O}(Jn^3 + 2^m J^m n^2 N)$. The memory requirements of storing the BLIS coefficients for N signals is are $\mathcal{O}(2^m J^m nN)$ (in addition to the memory costs of storing the wavelets). We also note the BLIS module is hand-crafted with no learnable parameters which means that these computations may be done offline as a preprocessing step.

Based on this analysis, a simple implementation of BLIS is linear with respect to the number of signals and therefore is well-suited to scale in the setting where there are many different signals defined on a single moderate-size network (which is the primary focus of this work and is often the case in the context of signal-level tasks).

It is also possible to modify our implementation to be scalable to large graphs. Tong et al. (2022) considered a modifies implentation of the wavlet transform which uses a diffusion module to compute $P\mathbf{x}, P^2\mathbf{x}, P^3\mathbf{x} \ldots, P^{2^J}\mathbf{x}$ via sparse matrix-vector multiplications and then compute the wavelets via vector-vector substraction. Notably, Wenkel et al. (2022) was able to use this method to achieve strong performance on large OGB benchmark data sets via a scattering-based network. If one implements BLIS in this manner, the computational cost is reduced to $O(2^{J+m}J^m(n + |E|))$ and the memory cost is reduced to $O(2^{m+J}J^m n)$ allowing for improved scalability.

I Models and Training

I.1 BLIS-Net architecture

A general and more complete description of the BLIS module and BLIS-Net architecture is given in Sections 3.1 and 3.4. For all BLIS-Net experiments, we utilize dyadic scales and choose $J = 4$ meaning that our $\mathcal{W}_J^{(1)}$ and $\mathcal{W}_J^{(2)}$ wavelet filter banks both contain six filters that we apply to our signal. Furthermore, we fix $m = 3$ meaning that we only utilize third-order coefficients. Our moment aggregation module utilizes first-order moments across the nodes. Our embedding layer and classification layer implented as a single, unified MLP, where the choice of hidden layers was determined from the data using 5-fold cross validation on the testing set. The hidden layer sizes are chosen from the set: $[(50,),(100,),(50,50),(150,50)]$. Dimensionality reduction is achieved with the linear layer to the first hidden layer, and the classification is performed with a layer that maps from the final hidden layer dimension to the number of classes. The MLP utilizes ReLU activations in between layers, Adam optimizer, an L2 regularization term of 0.01, and all other default settings on scikit-learn's implementation of the MLP classifier.

I.2 Graph Scattering Transform

For a complete description of the graph scattering transform, please refer to Section 2.5. For experiments involving the scattering transform, we utilize dyadic scales and choose $J = 4$ meaning that our $\mathcal{W}_J^{(1)}$ and $\mathcal{W}_J^{(2)}$ wavelet filter banks both contain five filters that we apply to our signal (because unlike in BLIS we don't use the low-pass). Following the convention of Gao et al. (2019) we utilize zeroth-, first-, and second-order scattering coefficients unless otherwise specified. To make a fair comparison with the BLIS module, we use all combination of scales in the second-order coefficients (Gao et al. (2019) only used $j_2 \geq j_1$) and when performing aggregation we only utilize the first moments. (Notably, one could readily modify the aggregation module in BLIS-Net to include higher moments as well.) The back-end MLP shares an identical construction to the one described in I.1 for the BLIS-Net architecture.

I.3 Baseline Graph Neural Networks

Graph Convolutional Network (GCN): The baseline GCN (Kipf and Welling, 2016) consists of two GCNConv layers, both followed by a ReLU activation function.

- The first GCNConv layer transforms the input features to a hidden dimension of 16.
- The second GCNConv layer maintains this dimension, mapping from 16 to 16.
- Following the convolutions, global mean pooling is applied to the node embeddings to obtain a graph-level representation.
- Finally, a linear layer is applied which outputs a dimension equal to the number of classes.

ChebNet: The baseline ChebNet model (Defferrard et al., 2016) is constructed using the same architecture as the above GCN but with the GCNConv layer replaced with a ChebConv layer with a filter size of 5.

Graph Isomorphism Network (GIN): The baseline GIN model (Xu et al., 2019) is structured with two GINConv layers, each of which is associated with its own MLP.

- The first GINConv layer utilizes an MLP that consists of two linear layers:
  1. The initial layer transforms the input features to a hidden dimension of 16.
  2. The subsequent layer retains this dimensionality, taking in the 16-dimensional space and outputting another 16-dimensional space. Between these two layers, a ReLU activation function is applied.
- The second GINConv layer has a similar MLP structure, mapping the 16-dimensional space from the first layer to another 16-dimensional space, again with a ReLU activation function in between.
- After both GINConv layers process the node features, a global mean pooling aggregates these features to produce a graph-level representation.
- This graph-level representation is then processed by a linear layer, transforming from 16 dimensions to a dimensionality equal to the number of classes.

Graph Attention Network (GAT): The GAT (Veličković et al., 2018) baseline is structured with two GATConv layers, each employing an attention mechanism.

- The first GATConv layer uses a single attention head, transforming the input features to a hidden dimension of 16.
- The second GATConv layer, operating in the same 16-dimensional space, continues this transformation, retaining the dimensionality of 16.
- An Exponential Linear Unit (ELU) activation function follows each of the convolutional layers.
- After the GATConv layers have processed the node features, a global mean pooling aggregates these features to produce a graph-level representation.

- This graph-level representation then undergoes a linear transformation, mapping from the 16-dimensional space to a dimensionality equal to the number of classes.

General, Powerful, Scalable (GPS) Graph Transformer : The GPS model (Rampášek et al., 2022) is designed to process graph-structured data with the incorporation of random walk-based positional encodings and the GPSConv layer.

Prior to feeding data into the model, random walk positional encodings of length 20 are added to the graph nodes.

- **Embeddings**: The input features undergo a linear transformation to produce node embeddings. Additionally, positional encodings are normalized and transformed into a space of dimension 8.
- **GPSConv Layers**: The architecture employs two GPSConv layers, each featuring:
  - A local message passing via GINEConv (Hu et al., 2020). This mechanism utilizes a two-layer MLP with hidden dimension 16 with ReLU activations.
  - Multi-head attention with 4 heads and an attention dropout rate of 0.5.
- **Classification**: Post the GPSConv processing, graph-level embeddings are obtained through a global addition pooling. These embeddings are directed into an MLP with 2 hidden layers to yield the final classification output.

GNNML1, GNNML3, PPGN: For these models, we utilize the default architecture, code, and parameters from the corresponding papers (Balcilar et al., 2021; Maron et al., 2019).

I.4 Training details

All data sets are subjected to a 70/30 train-test split, and performance metrics are computed by averaging results over a 5-fold cross-validation. For the baseline models, namely the Graph Convolutional Network (GCN), Graph Attention Network (GAT), Graph Isomorphism Network (GIN), and the general, powerful, scalable (GPS) graph Transformer, the Adam optimizer is employed with a learning rate of 0.01. These models are trained for 100 epochs to ensure convergence.

BLIS-Net is trained using the default sci-kit learn training protocol for the MLPClassifier, with full training details available in the documentation.

J Additional Descriptions of the data sets

Here we provide further descriptions of the data sets considered in our experiments and also provide summary statistics in Table 5.

J.1 Further details on the traffic data sets

To construct the signals from the PeMS data, we use the pre-processing procedure introduced in Guo et al. (2019). The graph structure is created by selecting sensors at least 3.5 miles apart and connecting adjacent sensors. Missing values in the graph signals are imputed using linear interpolation. PeMSD3, PeMSD4, PeMSD7, and PeMSD8 respectively consist of traffic data from California's 3rd, 4th, 7th, and 8th congressional districts and provide two months of consecutive traffic data collected between 2016 and 2018 depending on the data set. The PeMSD3 and PeMSD7 data sets we used contained a measurement of traffic flow at each sensor location. The PeMSD4 and PeMSD8 data sets used contain three types of measurements at each sensor location: total flow, average speed, and average occupancy. For these data sets, we pass each measurement into the BLIS module independently and then concatenate.

J.2 Further details on the Partly Cloudy data sets

The "Partly Cloudy" data set, sourced from Richardson et al. (2018), comprises MRI data captured from participants aged 3-12 years and adults as they watched the Disney Pixar animated film "Partly Cloudy". The

Data set statistics	$\lvert V\rvert$	$\lvert E\rvert$	Number of signals	Number of classes	Signal dimension	Number of sub-data sets
Partly Cloudy	39	113	168	3	1	155
Synthetic same μ	100	358*	400	2	1	5
Synthetic different μ	100	358*	400	2	1	5
PEMSD3	358	546	26208	24,7,4	1	1
PEMSD4	307	340	16992	24,7,4	3	1
PEMSD7	883	866	28224	24,7,4	1	1
PEMSD8	170	274	17856	24,7,4	3	1

Table 5: Summary of the Data sets mentioned in the paper. For the traffic data set, the list under number of classes is specified in the context of a particular task. 24 corresponds to the HOUR task, 7 corresponds to the DAY task, and 4 corresponds to the WEEK task. In the case of the Partly Cloudy data set, the number of sub-data sets corresponds to the number of participants for the experiment, and each participant shares the same underlying graph. For the synthetic data set, the number of sub-data sets reflects that 5 replicates were conducted for each task, meaning that 400 unique signals were generated on each of 5 random graphs. This is done to characterize the variation depending on the random generation on the graph. Due to this, the asterisk next to 358 for the number of edges is reflective of the mode of the number of edges for the 5 replicates.

data set's full title is "MRI data of 3-12 year old children and adults during viewing of a short animated film". The study involved 122 children and 33 adults. While undergoing the MRI scan, participants simply watched the film without any specific task.

The film is notable for portraying the characters' bodily sensations, such as pain, and their mental states. Movie frame annotations—categorizing them as positive, neutral, or negative in emotion—are derived from the labels in the repository of the paper Rieck et al. (2020).

For data preprocessing and Region of Interest (ROI) extraction, we utilized nilearn. We constructed a spatial connectivity graph from the ROI centroids, linking each centroid to its five closest neighbors. (We then symmetrize the graph by then setting $A_{i,j} = 1$ if either $A_{i,j} = 1$ or $A_{j,i} = 1$). We then applied temporal smoothing to the time series data for each node, using a Gaussian kernel convolution with a σ value of 1.75. Since fMRI data is extremely noisy, this temporal smoothing was critical for optimal model performance, as is explored in Table 12.

J.3 Further details on the Synthetic data set

We consider the functions and graph generation methods described in 4.1. We generate the nodes of the graph by sampling 100 points randomly from $[0,1]^2$ and connect each node to it's 5-nearest neighbors. As with the Partly Cloudy data, the graph is symmetrized if necessary. In total, 400 signals are generated per graph, with 200 signals corresponding to $f_j^{(1)}$ and 200 signals corresponding to $f_j^{(2)}$ to result in balanced classes. We generate 5 versions of this synthetic data set to control for randomness in the sampling of the vertices and the generation of signals.

K Ablation Study and Additional Experiments

BLIS-Net relies on pairing the BLIS module with an aggregation module, a dimension-reduction module (parametrized by an MLP), and a classification module (also parameterized by an MLP). However, one could also utilize the BLIS module in many other ways. In Tables 6, 7, 8, 9, 10, and 11, we show that the BLIS module can also be paired with shallow classifiers such as logistic regression (LR), random forest (RF), support vector classifier, and extreme gradient boosting (XGB) and also examine the performance of scattering with these same classifiers. (We also consider Scattering + MLP for direct comparison to BLIS-Net.) Notably, we perform these experiments on the PEMS04 and PEMS08 data sets in addition to those in the main body.

We see that BLIS-based methods generally perform well and consistently outperform the analogous scattering methods. For example, on the Partly Cloudy fMRI data set, BLIS-Net with the $\mathcal{W}_J^2$ wavelets and a simple logistic regression classifier is able to achieves 65.9% accuracy whereas the corresponding scattering implementation achieves only 53.1%. On the traffic data sets, we note that BLIS + XGB has the overall best performance, usually slightly better than BLIS-Net. On the synthetic data and the fMRI data, BLIS-Net is the top performer, followed by BLIS + logistic regression.

K.1 Denoising in the fMRI data set

fMRI data is extremely noisy. Therefore, in our experiments on the fMRI data, we performed a Gaussian smoothing over the time variable. Importantly, we note that we applied the same smoothing procedure for all methods. Results with and without the smoothing are shown in Table 12. We see that without the smoothing all methods perform poorly, with GPS being the top performing method at 42.0% followed closely by BLIS-Net (W2) at 41.5%. After the smoothing, the message passing networks (GCN, GAT, and GIN) continue to perform poorly (at most 42.1%). GPS improves from 42.0 to 56.4%, scattering improves from 40.3/40.7% to 60.6/62.3% and BLIS-Net improves from 41.1/41.5% to 67.1/68.3%.

PEMS03	HOUR	DAY	WEEK
GCN	27.8	14.1	30.8
GAT	36.4	23.5	30.8
GIN	14.0	14.3	30.8
GPS	57.4	49.6	31.9
ChebNet	58.8	**56.8**	**61.9**
GNNML1	6.8	15.2	30.3
GNNML3	61.5	49.2	36.3
PPGN	-	-	-
BLIS-Net (W1)	63.1	53.1	54.8
BLIS-Net (W2)	68.3	**56.3**	**61.7**
BLIS + LR (W1)	49.0	37.0	43.4
BLIS + LR (W2)	53.0	42.2	46.7
BLIS + RF (W1)	63.4	52.3	52.9
BLIS + RF (W2)	63.5	53.4	55.7
BLIS + SVC (W1)	49.1	35.1	37.9
BLIS + SVC (W2)	49.5	35.9	41.6
BLIS + XGB (W1)	**68.8**	54.0	52.6
BLIS + XGB (W2)	**69.2**	**56.3**	56.2
Scattering + MLP (W1)	58.2	45.6	46.4
Scattering + MLP (W2)	60.4	49.5	51.4
Scattering + LR (W1)	42.3	33.1	37.8
Scattering + LR (W2)	46.0	33.2	39.1
Scattering + RF (W1)	56.0	44.8	43.9
Scattering + RF (W2)	57.9	46.6	48.4
Scattering + SVC (W1)	43.5	28.3	32.6
Scattering + SVC (W2)	46.5	30.5	36.1
Scattering + XGB (W1)	56.7	42.8	41.6
Scattering + XGB (W2)	59.5	44.6	45.7

Table 6: Accuracy on the PEMS03 traffic data set. **Best** and **second** best results are colored.

PEMS04	HOUR	DAY	WEEK
GCN	38.1	19.5	28.6
GAT	43.3	23.5	30.8
GIN	39.8	17.7	28.6
GPS	66.5	67.0	31.7
ChebNet	67.7	56.6	64.2
GNNML1	5.4	15.8	27.1
GNNML3	59.8	48.0	43.9
PPGN	-	-	-
BLIS-Net (W1)	82.9	87.8	91.1
BLIS-Net (W2)	84.2	91.9	92.3
BLIS + LR (W1)	74.7	71.4	69.5
BLIS + LR (W2)	71.5	69.4	68.2
BLIS + RF (W1)	82.4	89.8	90.9
BLIS + RF (W2)	80.7	88.5	89.5
BLIS + SVC (W1)	71.9	75.5	77.0
BLIS + SVC (W2)	69.5	73.5	75.6
BLIS + XGB (W1)	**86.4**	**93.9**	**93.6**
BLIS + XGB (W2)	**86.1**	**92.8**	**92.9**
Scattering + MLP (W1)	78.5	83.2	83.8
Scattering + MLP (W2)	81.2	85.9	86.4
Scattering + LR (W1)	58.8	47.7	44.7
Scattering + LR (W2)	63.3	49.8	47.6
Scattering + RF (W1)	76.8	79.0	79.3
Scattering + RF (W2)	78.4	82.9	82.6
Scattering + SVC (W1)	60.0	55.9	55.6
Scattering + SVC (W2)	64.2	62.9	61.0
Scattering + XGB (W1)	81.3	79.1	75.8
Scattering + XGB (W2)	82.6	82.9	79.8

Table 7: Accuracy on the PEMS04 traffic data set.

PEMS07	HOUR	DAY	WEEK
GCN	27.4	14.6	28.5
GAT	33.2	22.2	36.5
GIN	14.3	15.8	28.4
GPS	39.9	27.7	30.4
ChebNet	54.0	61.4	72.9
GNNML1	5.4	15.8	27.1
GNNML3	59.8	48.0	43.9
PPGN	-	-	-
BLIS-Net (W1)	63.5	**72.9**	**76.8**
BLIS-Net (W2)	63.4	71.0	**77.3**
BLIS + LR (W1)	47.3	46.2	54.5
BLIS + LR (W2)	43.4	41.0	51.3
BLIS + RF (W1)	60.7	67.7	71.9
BLIS + RF (W2)	57.4	62.7	67.6
BLIS + SVC (W1)	51.1	53.7	56.7
BLIS + SVC (W2)	43.2	41.8	46.9
BLIS + XGB (W1)	**68.6**	**74.7**	75.3
BLIS + XGB (W2)	**64.8**	66.5	69.1
Scattering + MLP (W1)	54.0	53.3	56.9
Scattering + MLP (W2)	54.3	55.2	61.6
Scattering + LR (W1)	36.7	33.3	39.4
Scattering + LR (W2)	36.7	29.9	40.4
Scattering + RF (W1)	53.5	51.9	52.6
Scattering + RF (W2)	52.7	53.4	56.5
Scattering + SVC (W1)	39.7	35.5	38.7
Scattering + SVC (W2)	40.6	34.7	42.0
Scattering + XGB (W1)	53.2	50.2	49.1
Scattering + XGB (W2)	54.1	50.9	52.6

Table 8: Accuracy on the PEMS07 traffic data set.

PEMS08	HOUR	DAY	WEEK
GCN	33.3	20.4	32.3
GAT	38.0	28.3	35.3
GIN	24.5	14.5	32.1
GPS	67.7	67.9	62.3
ChebNet	61.9	69.4	75.4
GNNML1	4.5	14.5	28.9
GNNML3	60.2	58.1	49.6
PPGN	-	-	-
BLIS-Net (W1)	83.9	92.9	93.4
BLIS-Net (W2)	85.9	94.9	**95.6**
BLIS + LR (W1)	69.5	76.5	78.1
BLIS + LR (W2)	71.9	80.2	81.8
BLIS + RF (W1)	83.7	92.7	91.6
BLIS + RF (W2)	83.9	93.5	93.6
BLIS + SVC (W1)	72.4	85.2	84.9
BLIS + SVC (W2)	73.8	87.4	89.5
BLIS + XGB (W1)	**87.2**	**95.1**	94.7
BLIS + XGB (W2)	**87.7**	**96.0**	**96.1**
Scattering + MLP (W1)	81.0	89.9	89.3
Scattering + MLP (W2)	82.2	92.0	90.7
Scattering + LR (W1)	56.6	56.1	54.1
Scattering + LR (W2)	58.6	60.1	57.8
Scattering + RF (W1)	79.2	86.0	84.2
Scattering + RF (W2)	80.6	88.1	87.9
Scattering + SVC (W1)	63.4	71.5	66.2
Scattering + SVC (W2)	66.1	76.6	72.0
Scattering + XGB (W1)	81.8	87.2	81.9
Scattering + XGB (W2)	83.6	89.6	86.1

Table 9: Accuracy on the PEMS08 traffic data set.

Synthetic	Different μ	Same μ
GCN	99.0 ± 0.4	91.7 ± 2.0
GAT	99.2 ± 0.5	96.4 ± 0.6
GIN	99.5 ± 0.2	91.3 ± 1.4
GPS	95.4 ± 5.9	97.7 ± 0.9
ChebNet	99.1 ± 0.3	97.3 ± 0.6
GNNML1	99.3 ± 0.2	98.3 ± 0.4
GNNML3	99.8 ± 0.0	98.8 ± 0.4
PPGN	77.7 ± 9.9	62.4 ± 6.0
BLIS-Net (W1)	$\mathbf{100.0 \pm 0.0}$	97.7 ± 0.5
BLIS-Net (W2)	99.5 ± 0.3	$\mathbf{98.6 \pm 0.4}$
BLIS + LR (W1)	$\mathbf{100.0 \pm 0.0}$	98.5 ± 1.0
BLIS + LR (W2)	$\mathbf{100.0 \pm 0.0}$	$\mathbf{98.8 \pm 0.4}$
BLIS + RF (W1)	99.2 ± 0.4	97.7 ± 0.6
BLIS + RF (W2)	99.4 ± 0.1	97.1 ± 0.7
BLIS + SVC (W1)	99.4 ± 0.1	95.7 ± 1.5
BLIS + SVC (W2)	$\mathbf{100.0 \pm 0.0}$	95.5 ± 1.9
BLIS + XGB (W1)	99.5 ± 0.3	98.4 ± 0.1
BLIS + XGB (W2)	99.3 ± 0.0	97.7 ± 0.7
Scattering + MLP (W1)	97.7 ± 1.0	96.5 ± 1.2
Scattering + MLP (W2)	88.3 ± 4.3	96.8 ± 1.0
Scattering + LR (W1)	97.7 ± 0.7	96.1 ± 1.0
Scattering + LR (W2)	86.9 ± 4.9	95.3 ± 1.5
Scattering + RF (W1)	95.9 ± 1.6	94.5 ± 1.6
Scattering + RF (W2)	73.4 ± 8.0	94.1 ± 1.6
Scattering + SVC (W1)	98.1 ± 0.9	95.0 ± 1.5
Scattering + SVC (W2)	87.5 ± 4.9	93.5 ± 1.8
Scattering + XGB (W1)	95.2 ± 1.9	93.1 ± 2.5
Scattering + XGB (W2)	81.9 ± 7.2	94.7 ± 1.5

Table 10: Accuracy on the synthetic data sets.

Partly Cloudy	Emotion classification
GCN	39.3 ± 5.9
GAT	40.6 ± 6.1
GIN	42.1 ± 6.0
GPS	56.4 ± 4.3
ChebNet	50.2 ± 5.3
GNNML1	48.8 ± 5.4
GNNML3	45.2 ± 5.6
PPGN	42.0 ± 5.3
BLIS-Net (W1)	$\mathbf{67.1 \pm 4.3}$
BLIS-Net (W2)	$\underline{\mathbf{68.3 \pm 3.6}}$
BLIS + LR (W1)	62.4 ± 5.4
BLIS + LR (W2)	65.9 ± 5.2
BLIS + RF (W1)	61.5 ± 5.2
BLIS + RF (W2)	63.0 ± 4.5
BLIS + SVC (W1)	56.2 ± 5.3
BLIS + SVC (W2)	59.0 ± 5.0
BLIS + XGB (W1)	61.1 ± 5.7
BLIS + XGB (W2)	62.8 ± 5.1
Scattering + MLP (W1)	60.6 ± 4.9
Scattering + MLP (W2)	62.3 ± 5.1
Scattering + LR (W1)	51.2 ± 5.8
Scattering + LR (W2)	53.1 ± 5.9
Scattering + RF (W1)	56.1 ± 6.0
Scattering + RF (W2)	58.8 ± 5.5
Scattering + SVC (W1)	51.5 ± 5.9
Scattering + SVC (W2)	54.2 ± 6.1
Scattering + XGB (W1)	56.2 ± 6.0
Scattering + XGB (W2)	58.3 ± 5.8

Table 11: Accuracy on Partly Cloudy fMRI data.

Partly Cloudy	Emotion classification (No smoothing)	Emotion classification (with smoothing)
GCN	37.5 ± 4.9	39.3 ± 5.9
GAT	37.3 ± 4.8	40.6 ± 6.1
GIN	37.1 ± 4.5	42.1 ± 6.0
GPS	**42.0 ± 4.3**	56.4 ± 4.3
Scattering (W1)	40.3 ± 5.3	60.6 ± 4.9
Scattering (W2)	40.7 ± 5.8	62.3 ± 5.1
BLIS-Net (W1)	41.1 ± 5.0	**67.1 ± 4.3**
BLIS-Net (W2)	**41.5 ± 5.5**	**68.3 ± 3.6**

Table 12: Effect of Gaussian temporal smoothing on the accuracy on Partly Cloudy fMRI data.