Adaptive Gaussian Processes on Graphs via Spectral Graph Wavelets

Felix L. Opolka[*]
University of Cambridge
& Invenia Labs

Yin-Cong Zhi[*]
University of Oxford

Pietro Liò
University of Cambridge

Xiaowen Dong
University of Oxford

Abstract

Graph-based models require aggregating information in the graph from neighbourhoods of different sizes. In particular, when the data exhibit varying levels of smoothness on the graph, a multi-scale approach is required to capture the relevant information. In this work, we propose a Gaussian process model using spectral graph wavelets, which can naturally aggregate neighbourhood information at different scales. Through maximum likelihood optimisation of the model hyperparameters, the wavelets automatically adapt to the different frequencies in the data, and as a result our model goes beyond capturing low frequency information. We achieve scalability to larger graphs by using a spectrum-adaptive polynomial approximation of the filter function, which is designed to yield a low approximation error in dense areas of the graph spectrum. Synthetic and real-world experiments demonstrate the ability of our model to infer scales accurately and produce competitive performances against state-of-the-art models in graph-based learning tasks.

1 INTRODUCTION

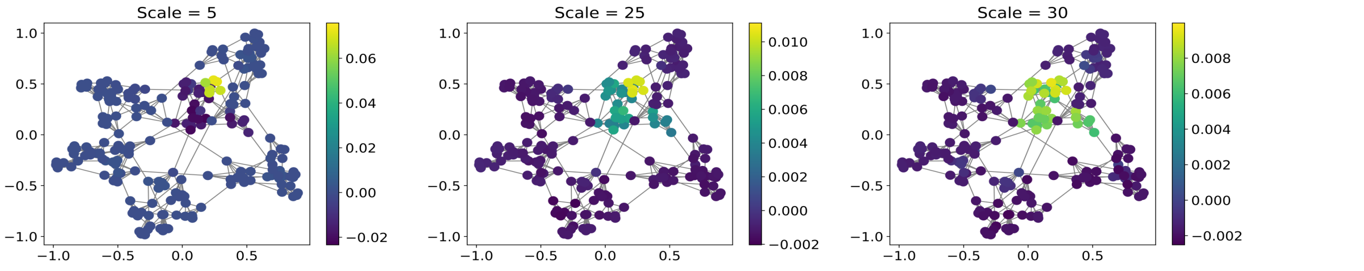
Many modern day data sets come in the form of graphs or networks, such as social networks, brain graphs, and protein-interaction networks, where additional information is represented by connective structures between data points. While node features on their own can be used by a variety of machine learning algorithms, the graph structure can often crucially enrich the model further. With the rise of graph signal processing (Shuman et al., 2013), graph neural networks (Wu et al., 2020), and geometric deep learning (Bronstein et al., 2021), there is now a rich library of tools to build models for graph structured data, making it possible to tackle a range of complex graph-based modelling tasks.

In particular, Gaussian process (GP) models are popular tools for taking into account the probabilistic nature of the data. A GP model on graphs would allow for modelling uncertainty associated with the nodes in the graph and making predictions on unlabelled nodes. A key requirement in building GPs on graphs is incorporating the graph information into the design of the GP kernel, for example using convolution-like operations (Ng et al., 2018; Walker and Glocker, 2019; Opolka and Liò, 2020; Li et al., 2020) or following the separable kernel design of multi-output GPs (Venkitaraman et al., 2020; Zhi et al., 2020).

The core consideration when incorporating graph structure into a model design is how much neighbours at varying distances should influence the prediction at a certain node. Early spectral approaches rely on the Fourier basis when designing graph-based operators (Bruna et al., 2014; Defferrard et al., 2016), which is fully localised in the frequency domain but not in the spatial domain, hence requiring a polynomial approximation of the graph Laplacian to enforce spatial localisation. We instead propose an approach using wavelets, which offer a natural way of trading off between spectral and spatial resolution—and thus localisation—in both domains. The degree of spatial localisation is implicitly controlled by a single wavelet scale parameter defined in the spectral domain (visualised in Figure 1), which makes graph wavelets a natural tool to enable a more flexible notion of neighbourhood of varying size. Moreover, the single scale parameter enables the model to adjust the effective neighbourhood sizes to the properties of the data when incorporated into a model that allows learning hyperparameters, such as a GP.

Beyond flexible control of neighbourhood size, using wavelets allows combining filters of different scales. Real-world networks such as connectivity patterns

Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151. Copyright 2022 by the author(s).



(a) (b) (c)

Figure 1: The Mexican Hat wavelet transform of a δ signal on the focal node. With different scales, the wavelet is able to capture different neighbourhood information weighted in a continuous manner.

in the brain or metabolic or social interactions networks often exhibit such multi-scale community structure (Ravasz and Barabási, 2003; Clauset et al., 2008; Dutkowski et al., 2013), where sets of densely connected nodes in turn form densely connected communities (see Figure 2a for a synthetic example). These graphs often naturally form the domain of multi-scale signals, which can be modelled through wavelets by combining filters of multiple scales. Figure 2b shows an example of how low-pass and band-pass filters are combined into a more complex wavelet filter, which then captures signal components varying at different scales (Figures 2c-e).

In this work, we introduce a novel graph GP model that uses spectral graph wavelets to incorporate graph structure into the GP kernel. Building on the convenient properties of the wavelet transform, the wavelet graph GP can naturally model continuous neighbourhoods of varying sizes and by extension multi-scale graph signals. The kernel filters are learnable such that their responses can adapt to the observed graph and data. To bypass the expensive eigen-decomposition of the graph Laplacian, we develop a fast approximation to the wavelet-based filtering, which still allows us to directly optimise the wavelet scales and reduces the approximation error on parts of the spectrum with most eigenvalues. We show that our approximation is more suitable for wavelet filters than the Chebyshev polynomial approximation commonly used for existing low-pass filtering approaches. Through experiments, we demonstrate accurate recovery of scales on a synthetic graph and evaluate our model on benchmark data sets, showing model performance is competitive against state-of-the-art graph-based models.

2 PRELIMINARIES

Gaussian Processes. Consider data of the form $(\mathbf{X}, \mathbf{y})$ where we have inputs $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^\top$ and labels $\mathbf{y} \in \mathbb{R}^N$, a GP f is defined as

$$f(\mathbf{x}) \sim \mathcal{GP}\big(m(\mathbf{x}), \mathcal{K}_\theta(\mathbf{x}, \mathbf{x}')\big) \tag{1}$$

for any inputs $\mathbf{x}, \mathbf{x}'$, where $m(\cdot)$ is the mean function, and $\mathcal{K}_\theta(\cdot, \cdot)$ is the symmetric and positive definite kernel function. GPs are Bayesian regression models known for the ability to incorporate prior information and having a closed form solution in computing the posterior. When predicting a new data point, the model provides both point predictions and confidence intervals. In addition, GPs provide a marginal log-likelihood on the training data, and in optimising this likelihood we can find the optimal hyperparameters based on the data.

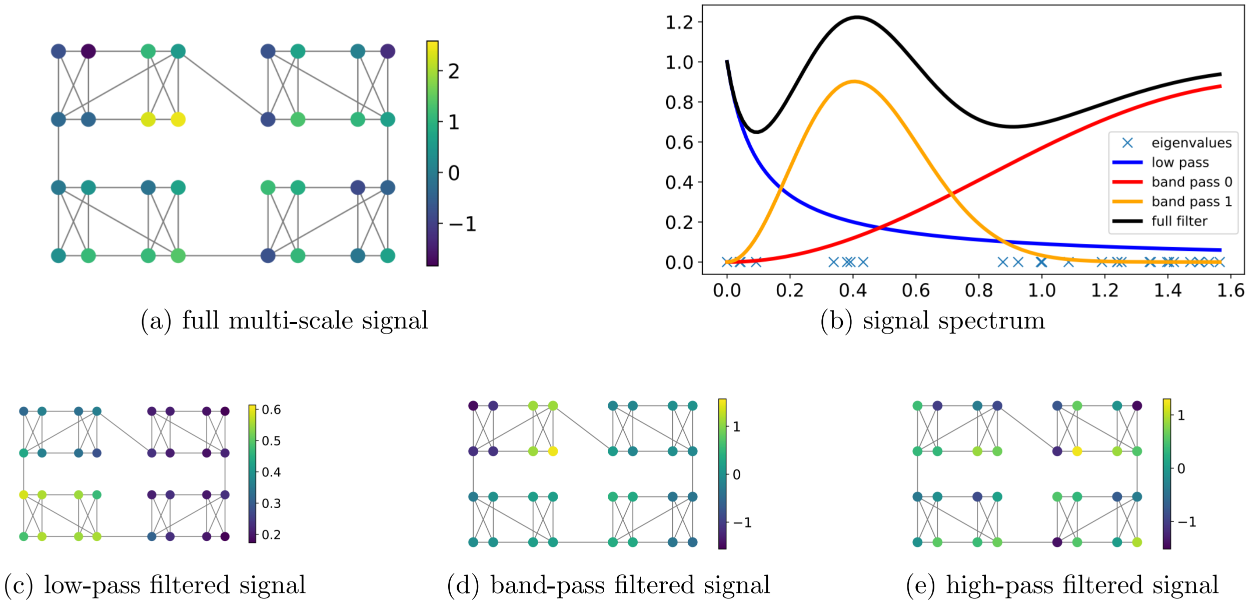
The limitations of GPs are in the inference step where two problems can arise. If a non-Gaussian likelihood is assumed on the data, for classification tasks for example, then the posterior will be analytically intractable. The inference step will also be problematic if the number of data N becomes large as it requires an expensive $\mathcal{O}(N^3)$ matrix inversion. Both problems can be addressed by approximating the posterior through a variational approach, here, a set of inducing points $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_M]^\top$ are introduced and form the inducing random variables $\mathbf{u} = [f(\mathbf{z}_1), \ldots, f(\mathbf{z}_M)]^\top$ that is a subset of the GP $f(\mathbf{x})$. Assuming the GP prior of $\mathbb{P}(\mathbf{u}) \sim \mathcal{N}(0, \mathbf{K_{zz}})$ where $[\mathbf{K_{zz}}]_{ij} = \mathcal{K}_\theta(\mathbf{z}_i, \mathbf{z}_j)$, the conditional GP has the following distribution

$$f(\mathbf{x})|\mathbf{u} \sim \mathcal{GP}(\mathbf{k}_{\mathbf{zx}}^\top \mathbf{K}_{\mathbf{zz}}^{-1} \mathbf{u}\ ,\ \mathcal{K}(\mathbf{x}, \mathbf{x}) - \mathbf{k}_{\mathbf{zx}}^\top \mathbf{K}_{\mathbf{zz}}^{-1} \mathbf{k}_{\mathbf{zx}}) \tag{2}$$

where $\mathbf{k_{zx}}$ are the cross covariances $[\mathcal{K}(\mathbf{z}_1, \mathbf{x}), \ldots, \mathcal{K}(\mathbf{z}_M, \mathbf{x})]^\top$. The variational posterior distribution $q(\mathbf{u})$ is assumed to be a multivariate Gaussian with mean $\mathbf{m}$ and covariance matrix $\mathbf{S}$ to be found through maximising the Evidence Lower Bound (ELBO)

$$\mathcal{L}(\theta, \mathbf{Z}, \mathbf{m}, \mathbf{S}) = \sum_{n=1}^{N} \mathbb{E}_{q(f(\mathbf{x}_n))}[\log \mathbb{P}(y_n | f(\mathbf{x}_n))] \tag{3}$$

$$- \mathrm{KL}[q(\mathbf{u}||\mathbb{P}(\mathbf{u})]. \tag{4}$$



(a) full multi-scale signal

(b) signal spectrum

(c) low-pass filtered signal

(d) band-pass filtered signal

(e) high-pass filtered signal

Figure 2: Visualisation of how wavelet filters can be used to capture multi-scale properties in both the graph structure and the graph signal. Figure (a) shows a graph with two levels of clusters (4-node clusters and 8-node clusters). These clusters are reflected in the gaps (around 0.2 and 0.6) in the spectrum of the graph in Figure (b). The signal is obtained by filtering a random signal with the filter in (b), purposefully highlighting the three eigenvalue clusters. Figures (c) - (e) show how the full signal from (a) decomposes into the three filter components. As expected, the low-pass signal varies mostly on the highest cluster level (between 8-node clusters), the band-pass signal mostly on the second cluster level (between 4-node clusters), and the high-pass signal from node to node.

Typically the parameters are optimised via stochastic gradient descent. We refer readers to (Rasmussen and Williams, 2005) for a more comprehensive overview.

Spectral Filtering and Wavelets on Graphs. We refer to the filtering of a signal as the process of highlighting specific frequency components in the signal while de-emphasising others with the aim of obtaining a function more suitable for the prediction task. Let $\mathcal{G} = (\mathcal{V}, \mathbf{A})$ be a graph with vertex set $\mathcal{V} = \{v_1, \ldots, v_N\}$ and adjacency matrix $\mathbf{A}$, we define the notion of spectral filtering on graphs (Shuman et al., 2013) based on the graph Laplacian defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{D}$ is the diagonal degree matrix. Additionally, the commonly used normalised graph Laplacian is computed as $\tilde{\mathbf{L}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}}$. This is often preferred due to the boundedness of its eigenvalues to the interval $[0, 2]$ and the scaling of the graph edge weights (Shuman et al., 2013), hence our model will make use of this normalised version throughout.

Assuming that $\mathcal{G}$ is undirected, the Laplacian is symmetric and admits the eigen-decomposition $\tilde{\mathbf{L}} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ where $\mathbf{U}$ contains the eigenvectors and $\mathbf{\Lambda}$ is the diagonal matrix of eigenvalues. For any function $\mathbf{f}$ on $\mathcal{G}$ (or graph signal), the graph Fourier transform is defined as $\mathbf{U}^\top\mathbf{f}$ and computes the amplitude of each eigenvector in the function $\mathbf{f}$. Filtering on graphs is then achieved in the graph spectral domain by applying a function on the eigenvalues as $g(\mathbf{\Lambda})$, and we write $\hat{\mathbf{f}} = \mathbf{U}g(\mathbf{\Lambda})\mathbf{U}^\top\mathbf{f}$ as the filtered signal (or function), and $\mathbf{U}g(\mathbf{\Lambda})\mathbf{U}^\top$ is referred to as the graph filtering matrix.

The graph Fourier transform $\mathbf{U}$ is localised in the graph spectral domain as each eigenvector only contributes a single frequency to the construction of $\mathbf{f}$. However, they are not localised in space as each eigenvector of $\mathbf{f}$ is on the entire spatial domain. Wavelet transform addresses this issue by decomposing a function $\mathbf{f}$ into a linear combination of basis function that are both localised in space and frequency. The definition of graph wavelets is derived from spectral graph theory by Hammond et al. (2011b) and will form the basis of the wavelets we utilise. The transform is an operator function of the graph Laplacian determined by a function g as follows: $b_\beta(\tilde{\mathbf{L}}) = \mathbf{U}g(\beta\mathbf{\Lambda})\mathbf{U}^\top$. The function g is applied in the graph spectral domain, but spatially it will also be localised if chosen from the library of mother wavelets. The scale parameter β then plays the role of controlling the localisation of the transform. We make use of the Mexican Hat wavelet, which we will present later on along with our model formulation. The spatial localisation can be demonstrated by applying the wavelet transform to an impulse signal on the graph

$b_\beta(\tilde{\mathbf{L}})\delta_n$, where $\delta_n = 1$ at node n and 0 elsewhere. This is presented in Figure 1 where the various scales β lead to different proximity of neighbourhoods. For each scale, the different hop neighbourhoods are also weighted in a continuous manner that decays to 0 once far enough away from the centre node. This allows the aggregation to happen in a non-linear manner to extract additional information for each node.

3 METHODOLOGY

Graph wavelet GP. We describe a Gaussian process model for the task of semi-supervised node-level prediction on a graph $\mathcal{G} = (\mathcal{V}, \mathbf{A})$ with N nodes. The nodes of the graph are commonly associated with a set of features $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, which form the feature matrix $\mathbf{X} \in \mathbb{R}^{N \times K}$. As we have seen in Section 2, given a graph signal $\mathbf{f} \in \mathbb{R}^N$ on the graph domain, we can apply a wavelet filter $g_\theta(\lambda)$ with scale parameters θ as follows:

$$\hat{\mathbf{f}} = \mathbf{U} g_\theta(\mathbf{\Lambda}) \mathbf{U}^\top \mathbf{f}, \tag{5}$$

where $\mathbf{U}$ and $\mathbf{\Lambda}$ are the eigenvectors and eigenvalues of the graph Laplacian of $\mathcal{G}$ such that $\tilde{\mathbf{L}} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top \in \mathbb{R}^{N \times N}$ and $\mathbf{\Lambda}$ is a diagonal matrix. The wavelet filter g_θ is applied element-wise to Λ. For brevity, we define $\mathbf{W}_\theta \coloneqq \mathbf{U} g_\theta(\mathbf{\Lambda}) \mathbf{U}^\top$ and refer to it as the *wavelet filter matrix.*

For the sake of conducting Bayesian inference, we assign a Gaussian process prior to the function $\mathbf{f}$

$$\mathbf{f} \sim \mathcal{GP}\left(m(\mathbf{x}), \mathcal{K}_\psi(\mathbf{x}, \mathbf{x}')\right), \tag{6}$$

with the mean function m and kernel function $\mathcal{K}_\psi$ with parameters ψ operating on the node features. On domains described by graphs with a finite number of nodes this prior is equivalent to a multivariate normal distribution with mean $\mathbf{m} = m(\mathbf{X}) \in \mathbb{R}^N$ and covariance $\mathbf{K} = \mathcal{K}(\mathbf{X}, \mathbf{X}) \in \mathbb{R}^{N \times N}$ respectively. As the wavelet filtering described in Equation 5 is a linear operation the filtered signal $\hat{\mathbf{f}}$ follows a Gaussian process prior

$$\hat{\mathbf{f}} \sim \mathcal{GP}\left(\mathbf{W}_\theta \mathbf{m}, \mathbf{W}_\theta \mathbf{K} \mathbf{W}_\theta^\top\right). \tag{7}$$

When combined with a likelihood $p(\mathbf{y} \,|\, \hat{\mathbf{f}})$, the model is capable of Bayesian inference of an output signal $\mathbf{y} \in \mathbb{R}^N$ by computing the posterior distribution $p(\hat{\mathbf{f}} \,|\, \mathbf{y})$. In case of regression, the likelihood is commonly assumed to be a normal distribution $p(\mathbf{y} \,|\, \hat{\mathbf{f}}) = \mathcal{N}(\mathbf{y} \,|\, \hat{\mathbf{f}}, \sigma^2 \mathbf{I})$ with observation noise σ^2 and the posterior distribution can be computed in closed form (Rasmussen and Williams, 2005). In case of classification, a categorical likelihood is assumed, leading to an intractable posterior. We then opt to approximate it with a variational posterior $q(\hat{\mathbf{f}})$ following Hensman et al. (2015).

Adaptive GP via learning wavelet scales. A key part of model design is the choice of wavelet filter g_θ (cf. Equation 5). A wide variety of mother wavelet functions are available, here, we choose the Mexican Hat wavelet function for the band-pass filters, defined as

$$b_\beta(\lambda) = \frac{2\sqrt{2}}{\sqrt{3}\pi^{\frac{1}{4}}} \left(\frac{\lambda}{\beta}\right)^2 \exp\left(-\frac{1}{2}\left(\frac{\lambda}{\beta}\right)^2\right) \tag{8}$$

with scale β. A band-pass filters emphasises the frequencies in an interval (or band) of the spectral domain. The location of that interval is controlled by the scale β, which thereby controls the localisation of the transform in the spatial and frequency domain. To model lower frequencies of the signal we choose a scaling function with a relatively fast decay as the low-pass filter, defined as

$$h_\alpha(\lambda) = \frac{1}{1 + \alpha\lambda} \tag{9}$$

with scale α. A low-pass filter emphasises the lower frequencies of a signal, corresponding to its smoother components, where smoothness is measured by the Dirichlet energy $\|\mathbf{f}\|_{\mathcal{G}} = \mathbf{f}^\top \mathbf{L} \mathbf{f}$. The scale α controls how much the filter smooths the signal. To obtain the combined effect of the low-pass and all band-pass filters, we can compute a full filter function as the sum of the individual filters. For L scales, this leads to the spectral filter function

$$g_\theta(\lambda) = h_\alpha(\lambda) + \sum_{l=1}^{L} b_{\beta_l}(\lambda) \tag{10}$$

with $\theta = \{\alpha, \beta_1, \ldots, \beta_L\}$, which is used to compute the wavelet filter matrix $\mathbf{W}_\theta = \mathbf{U} g_\theta(\mathbf{\Lambda}) \mathbf{U}^\top$, where the subscript highlights the dependence of the filter matrix on the scale parameters.

When the wavelet filter is applied to the GP prior as in Equation 7, the scale parameters θ can be treated as kernel hyperparameters and can be optimised as part of the model fitting process. This is achieved by maximising the marginal log-likelihood $p(\mathbf{y} \,|\, \theta, \psi)$ with respect to both the scale parameters θ and the parameters ψ of the node feature kernel $\mathcal{K}_\psi$ (cf. Equation 6):

$$\begin{aligned} \theta, \psi &= \arg\max_{\theta,\psi} p(\mathbf{y} \,|\, \theta, \psi) \\ &= \arg\max_{\theta,\psi} \int p(\mathbf{y} \,|\, \hat{\mathbf{f}}) p(\hat{\mathbf{f}} \,|\, \theta, \psi) \,\mathrm{d}\hat{\mathbf{f}}, \end{aligned} \tag{11}$$

where we highlight the dependence of the GP prior $p(\hat{\mathbf{f}} \,|\, \theta, \psi)$ on the hyperparameters by explicitly conditioning on them. In the case of classification, which prescribes a non-Gaussian likelihood, the marginal likelihood is intractable and we therefore resort to maximising a variational lower bound (Equation 4) on the marginal likelihood, again following Hensman et al.

(2015). This setup enables the model to learn to emphasise frequencies in the data that best describe the output signal $\mathbf{y}$ at hand. In Section 5, we examine the model's ability to recover the correct scale in a synthetic data experiment.

Spectrum-adaptive polynomial approximation. The model formulation described in previous sections requires computing the eigen-decomposition of the Laplacian of the input graph $\mathcal{G}$, which has computational complexity in $\mathcal{O}(N^3)$ and is therefore intractable for larger graphs. To alleviate this limitation, we opt for choosing to approximate the wavelet filter $g_\theta(\lambda)$ with a polynomial $p_\theta(\lambda) = \gamma_0 + \gamma_1\lambda + \ldots + \gamma_K\lambda^K \approx g_\theta(\lambda)$ of degree K, as previously suggested by Hammond et al. (2011b). This allows rewriting the filtering operation in Equation 5 as

$$\hat{\mathbf{f}} = \mathbf{U}g_\theta(\mathbf{\Lambda})\mathbf{U}^\top\mathbf{f} \approx \mathbf{U}p_\theta(\mathbf{\Lambda})\mathbf{U}^\top\mathbf{f} = p_\theta(\tilde{\mathbf{L}})\mathbf{f}. \tag{12}$$

This formulation circumvents the expensive eigendecomposition of the graph Laplacian and furthermore allows exploiting the sparsity of the Laplacian by using sparse matrix-vector multiplication to compute $p_\theta(\tilde{\mathbf{L}})\mathbf{f}$, which reduces the complexity of the filtering operation to $\mathcal{O}(KE)$, where E is the number of edges in the graph. Existing approaches have relied on a truncated Chebyshev polynomial approximation of the filtering operation and freely optimising the polynomial coefficients $\boldsymbol{\gamma} \in \mathbb{R}^{K+1}$ (Hammond et al., 2011b; Defferrard et al., 2016). In contrast, our approach is based on optimising the scale parameters (see previous sections) and we therefore require a polynomial approximation that is parameterised by the wavelet scales θ. A natural choice is the least squares approximation to the filter function $g_\theta(\lambda)$

$$\boldsymbol{\gamma}_\theta = (\mathbf{V}_\xi^\top\mathbf{V}_\xi)^{-1}\mathbf{V}_\xi^\top g_\theta(\boldsymbol{\xi}), \tag{13}$$

where $\boldsymbol{\xi} \in \mathbb{R}^S$ is a set $\{\xi_i\}_{i=1}^S$ of linearly spaced points on the spectral domain in the interval $[0, 2]$ and $\mathbf{V}_\xi \in \mathbb{R}^{S\times(K+1)}$ is the Vandermonde matrix for $\boldsymbol{\xi}$ up to degree K.

The above least-squares approximation minimises the approximation error uniformly on the spectral domain. However, in graphs with multi-scale characteristics, the eigenvalues are not uniformly distributed on the spectral domain but rather display *spectral gaps* corresponding to the different scales in the data (cf. Figure 2). As the filter function $g_\theta(\lambda)$ is only ever evaluated at the eigenvalues of the graph, a high approximation error of the polynomial approximation at those spectral gaps can be accepted in turn for a lower approximation error on parts of the spectrum with a higher density of eigenvalues. Following the ideas of Shuman et al. (2015); Fan et al. (2020), we achieve this by computing a weighted least square approximation of the filter function $g_\theta(\lambda)$, where the weights are chosen to be proportional to the spectral density of the graph (Mieghem, 2011, Chapter 6), which is defined as

$$p_\lambda(z) \coloneqq \frac{1}{N}\sum_{l=1}^{N} \mathbb{1}_{\{\lambda_l = z\}}. \tag{14}$$

Spectral density estimation aims to approximate this function without performing the expensive eigendecomposition of the graph Laplacian. We opt to employ the Kernel Polynomial Method (Lin et al., 2016; Li et al., 2019; Silver and Röder, 1994; Silver et al., 1996; Wang, 1994) to find an estimate of the spectral density function by first finding an estimate for the cumulative spectral density function $P_\lambda(z) \coloneqq \frac{1}{N}\sum_{l=1}^{N} \mathbb{1}_{\{\lambda_l \leq z\}}$. For each ξ_i from the set $\{\xi_i\}_{i=1}^S$ of S linearly spaced points on the spectral domain, we aim to find the number of eigenvalues less than or equal to ξ_i. This can be achieved via stochastic trace estimation (Girard, 1989), which provides us with a randomized algorithm for computing the trace of a matrix $\mathbf{B}$ and we use the Gaussian estimator

$$\operatorname{tr}(\mathbf{B}) = \mathbb{E}\left[\mathbf{z}^\top\mathbf{B}\mathbf{z}\right] \approx \frac{1}{R}\sum_{r=1}^{R}\mathbf{z}^\top\mathbf{B}\mathbf{z}, \qquad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{15}$$

where R is the number of Monte Carlo samples drawn for approximating the expectation. We thus require a matrix function Θ_{ξ_i} that maps the Laplacian $\tilde{\mathbf{L}}$ to a matrix whose trace equals the number of eigenvalues less or equal to ξ_i. This mapping is trivially given by $\Theta_{\xi_i}(\lambda) = \mathbb{1}_{\{\lambda \leq \xi_i\}}$. While we are not able to cheaply compute Θ_{ξ_i} exactly, we can approximate it using a Jackson-Chebyshev polynomial $\tilde{\Theta}_{\xi_i}$ (details of this approximation can be found in (Di Napoli et al., 2016; Puy and Pérez, 2018)). We obtain an approximation $\tilde{P}_\lambda(z)$ to the cumulative spectral density function by interpolating between the estimates at points ξ_i using monotonic piece-wise cubic interpolation $\mathcal{I}$

$$\tilde{P}_\lambda(z) = \mathcal{I}\left(\left\{\left(\xi_i, \frac{1}{N}\left[\frac{1}{R}\sum_{r=1}^{R}\mathbf{z}_r^\top\tilde{\Theta}_{\xi_i}(\tilde{\mathbf{L}})\mathbf{z}_r\right]\right)\right\}_{i=1}^{S}\right), \qquad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

Finally, differentiating $\tilde{P}_\lambda(z)$ with respect to z gives an approximation $\tilde{p}_\lambda(z)$ to the spectral density.

Using this estimate of the spectral density, we can compute weights $\boldsymbol{\omega} \in \mathbb{R}^S$ for each of the S sample points ξ_i on the spectral domain. We can then compute the weighted least squares coefficients

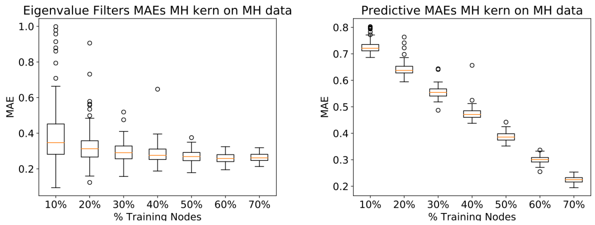
$$\boldsymbol{\gamma}_\theta = \underbrace{\left(\mathbf{V}^\top\operatorname{diag}(\boldsymbol{\omega})\mathbf{V}\right)^{-1}\mathbf{V}^\top\operatorname{diag}(\boldsymbol{\omega})}_{\text{projection matrix } \mathbf{P}} g_\theta(\boldsymbol{\xi}) \tag{16}$$

to be used in the polynomial approximation $p_\theta(\tilde{\mathbf{L}})$. The spectral density weights $\boldsymbol{\omega}$ may be pre-computed before training and combined into the projection matrix $\mathbf{P} \in \mathbb{R}^{(K+1)\times S}$, which projects from the exact filter values $g_\theta(\boldsymbol{\xi})$ to the polynomial coefficients $\boldsymbol{\gamma}_\theta$. Finally, these coefficients are used to approximate the wavelet filter matrix $\mathbf{W}_\theta \approx \gamma_0\mathbf{I} + \gamma_1\tilde{\mathbf{L}} + \gamma_2\tilde{\mathbf{L}}^2 + \ldots$, where we have dropped the coefficient's explicit dependence on the scale parameters θ for notational clarity.

4 RELATED WORK

Our work is first related to recent developments in developing GP models to handle graph-structured data, where the main challenge is to incorporate the graph information into the design of GP kernels. The first option is to directly encode the relational structure of nodes provided by the graph as an aggregation of the kernel matrix (Ng et al., 2018; Li et al., 2020; Liu et al., 2020; Cheng et al., 2020). A second option is to leverage notions of graph convolutions for the same purpose (Opolka and Liò, 2020; Walker and Glocker, 2019). From a slightly different perspective, the studies by Venkitaraman et al. (2020); Zhi et al. (2020) follow the literature of multi-output GPs with a separable kernel design. Finally, a Matérn GP on graphs has been proposed by Borovitskiy et al. (2021), although their model resembles kernels on graphs (Smola and Kondor, 2003). All of these studies, however, do not exploit the topological properties of the graph on which the GP is built; furthermore, in the context of graph GPs, only the recent work by Zhi et al. (2020) has attempted to learn an adaptive graph filter via a polynomial design, and their work only focuses on the vector-output setting. Our work proposes an adaptive GP that utilises the spectral graph wavelets to adapt to the multi-scale properties of the graph domain as well as the data it supports. The resulting kernel is semi-supervised for scalar-output GPs, but the graph wavelet can easily be adapted to vector-outputs.

Our study is more broadly addressing the recent attempts in incorporating signal processing concepts and tools into the design of graph-based learning models, especially the graph neural networks (GNNs) (Wu et al., 2020). One well-documented issue of these models is over-smoothing (Li et al., 2018; Oono and Suzuki, 2020) which, from a signal processing perspective, may be interpreted as a result of merely low-pass filtering of the graph signals (Wu et al., 2019). As a consequence, they may also not be suitable for scenarios where the labels exhibit a low level of homophily (Zhu et al., 2020). Several recent studies have attempted to address these issues by designing filters that go beyond low-frequency information (Min et al., 2020; Bo et al., 2021; Zheng et al., 2021; Liao et al., 2019; NT et al., 2020; Dong et al., 2021; Mingguo et al., 2021). Although these frameworks offer the possibility to go beyond low-pass filtering, they are all neural network models which typically require a large amount of training data and lack a measure of predictive uncertainty. Our work proposes a GP model that address both limitations, where the learnable wavelet filters offer the flexibility in representing different types of signal spectra.



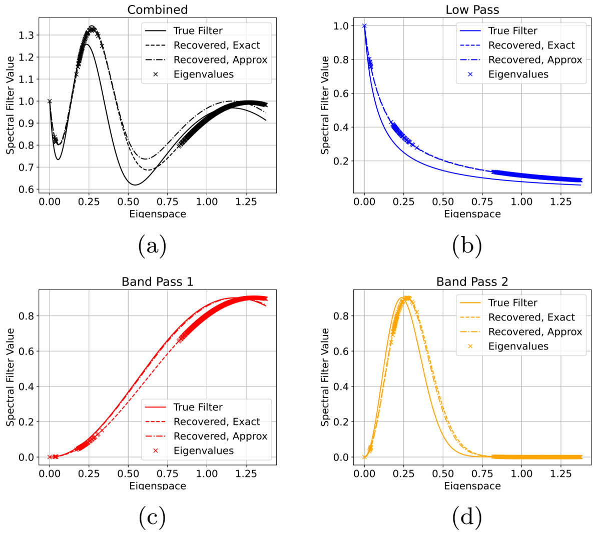
(a) (b)

Figure 3: (a) MAE between the recovered wavelet filter against ground truth at the eigenvalues. (b) MAE between predicted values at testing nodes and ground truth labels

Our work is finally related to multi-scale analysis of graph data. Various wavelet transforms have been developed to analyse both the graph and the signals at different scales (Coifman and Maggioni, 2006; Hammond et al., 2011a; Gama et al., 2019). For example, the work by Tremblay and Borgnat (2014) has adopted the spectral graph wavelets and tackled the problem of detecting community structure at multiple levels, while the application of wavelets by Xu et al. (2019) is used as an alternative to the graph Fourier basis. In the latter, the bases are however limited to low-pass and as a result lacks the spectral multi-scale property of our model. In terms of data defined on graphs, recent studies have utilised scattering transforms, which are based on diffusion wavelets, for applications such as node and graph classification as well as dimensionality reduction (Gao et al., 2019; Min et al., 2020). To the best of our knowledge, our framework is the first that incorporates graph wavelets into GP design for the same purpose.

5 EXPERIMENTS

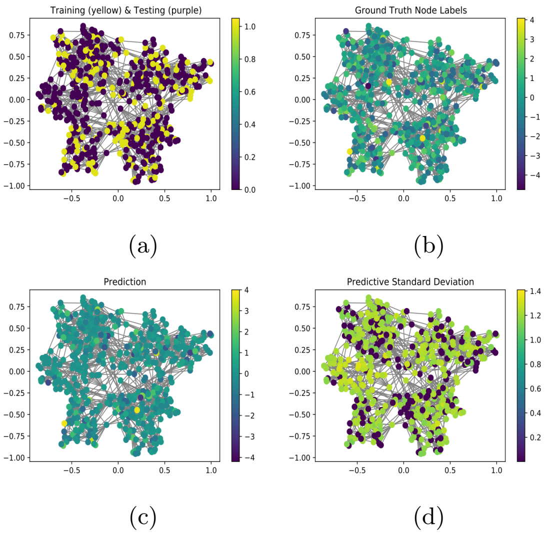
Synthetic Multi-Scale Graphs For Scales Recovery and Predictions. The concept of multi-scale corresponds to different things depending on if we are in the graph spatial or spectral domain. In the spectral domain, this is characterised by different dilation of the band pass filter, whereas spatially we often associate higher level scales as clustering of clusters. If the graph is spatially multi-scale, the different levels of clusters



(a) (b)

(c) (d)

Figure 4: Scale recoveries using exact wavelets and polynomial approximations on 50% of data. The ground truth (a) is made of a low-pass $a = 12$ and two band-passes $s = 1.2$ & 6 shown in (b)-(d).



(a) (b)

(c) (d)

Figure 5: (a) graph split into training (yellow) and testing (purple) nodes, only training node labels are made available to the model. (b) node labels of full graph. (c) prediction of full signal using only the training nodes. (d) Node standard deviation of posterior (these are 0 at training nodes).

translate to gaps in the eigenvalues, which means we can apply certain characteristics to each level by adjusting a number of ground truth wavelet filters. In this synthetic setting we apply our model to graph data simulated to have both spectral and spatial multi-scale properties. We show that in optimising the GP prior for the model fitting process, we can accurately recover different scales in the wavelets of the ground truth.

We start by sampling a multi-scale graph through a specially designed algorithm. We use the Erdős-Rényi (ER) random graph as the base generator, and the algorithm involves repeatedly sampling ER graphs to replace the nodes in that level. Continuous labels are then generated for the nodes by sampling from a Gaussian prior with wavelets in the kernels. Let $\mathbf{W}_\phi$ represent a set of wavelets with pre-chosen set of scales $\phi = \{a = 12, s_1 = 1.2, s_2 = 6\}$ such that $\mathbf{W}_\phi = h_a(\tilde{\mathbf{L}}) + g_{s_1}(\tilde{\mathbf{L}}) + g_{s_2}(\tilde{\mathbf{L}})$. We do not specify any node attributes, hence an identity kernel is assumed for $\mathbf{K}$. To obtain the node labels, we sample from the Gaussian process

$$\mathbf{y} \sim \mathcal{GP}(0, \mathbf{W}_\phi \mathbf{W}_\phi^\top). \tag{17}$$

We split the labels $\mathbf{y}$ randomly into $\mathbf{y}_{\text{train}}$ and $\mathbf{y}_{\text{test}}$, with only $\mathbf{y}_{\text{train}}$ made available to the model for training. The model we use will take the form $\mathbf{f} \sim \mathcal{GP}(0, \mathbf{W}_\theta \mathbf{W}_\theta^\top)$ where $\mathbf{f}$ is the prior between the training and testing nodes and $\theta = \{\alpha, \beta_1, \beta_2\}$ are parameters to be found based on the training labels provided. As in the semi-supervised setting, the full graph will be made available to the model through computing the full $\mathbf{W}_\theta$ matrix, and θ is then found by maximising the marginal log-likelihood $\mathbb{P}(\mathbf{y}_{\text{train}}|\theta, \mathcal{G})$. Once the hyperparameters are found we can condition on the training data to obtain the predictive distribution $\mathbb{P}(\mathbf{y}_{\text{test}}|\mathbf{y}_{\text{train}}, \theta, \mathcal{G})$. This distribution provides us with the mean prediction and confidence intervals as shown in Figure 5.

We look at two particular performance measures: the mean absolute error (MAE) between the ground truth wavelet filter and the recovered filter at the eigenvalues, and the MAE between $\mathbf{y}_{\text{test}}$ and the posterior mean of $\mathbb{P}(\mathbf{y}_{\text{test}}|\mathbf{y}_{\text{train}}, \theta, \mathcal{G})$. For each selection and percentage of nodes used during training, we sample multiple labels as in (17) to recover the filters from. The MAEs can be found in Figure 3. The ground truth and recovered filters (via both exact formulation and approximation) for one specific example are presented in Figure 4a, while the individual filters h_a, g_{s_1}, g_{s_2} are also shown in Figure 4b - 4d. More results on scale recovery and predictions on synthetic data (including comparison against baselines) are presented in the Appendix.

Semi-Supervised Classification on Graphs. We apply Wavelet Graph GP (WGGP) to three citation networks (Sen et al., 2008), which are commonly used as benchmark data sets for graph-based models. Here, the underlying graph consists of citations and the node features are bag-of-words (BOW) re-weighed using the popular term frequency-inverse document frequency

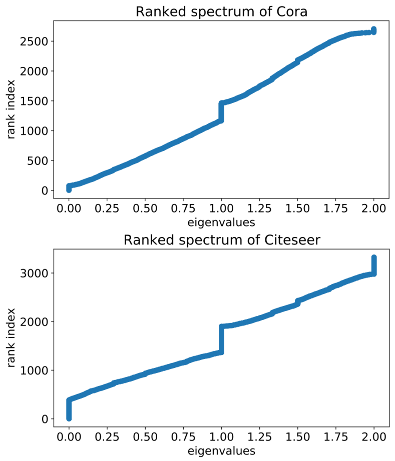
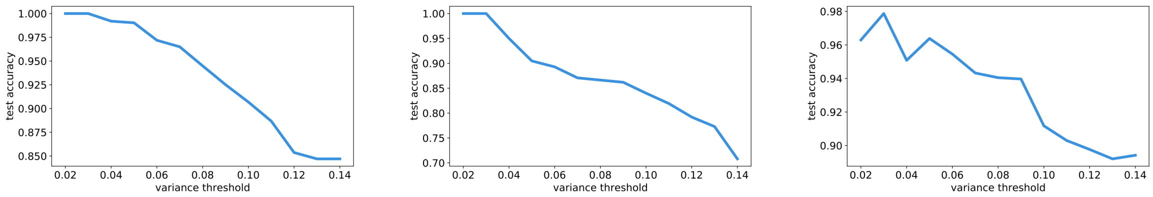


Figure 6: Ranked spectra of Cora and Citeseer. Both present distinct ranges of eigenvalues, suggesting multi-scale graph structure.

Method	Cora	Citeseer	PubMed
GCN (Kipf and Welling, 2017)	80.5 ±0.8	68.1 ±1.3	77.8 ±0.7
GAT (Veličković et al., 2018)	82.6 ±0.7	72.2 ±0.9	76.7 ±0.5
ChebNet (Defferrard et al., 2016)	78.0 ±1.2	70.1 ±0.8	69.8 ±1.1
LanczosNet (Liao et al., 2019)	79.5 ±1.8	66.2 ±1.9	78.3 ±0.3
AdaLanczosNet (Liao et al., 2019)	80.4 ±1.1	68.7 ±1.0	78.1 ±0.4
GP (Ng et al., 2018)	60.8	54.7	71.5
GGP (Ng et al., 2018)	80.9	69.7	77.1
GGP-X (Ng et al., 2018)	84.7	75.6	82.4
ChebGP (ours)	79.7	66.5	77.2
WGGP (ours)	84.7	70.8	78.4
WGGP-X (ours)	87.5	76.8	90.0

Table 1: Predictive accuracies of our proposed Wavelet Graph Gaussian Process model compared to a number of baselines. Results are reported with the mean and standard deviation over 10 runs except for Gaussian process models, which do not require random weight initialisations.



(a) Cora

(b) Citeseer

(c) Pubmed

Figure 7: We evaluate the performance of our WGGP model when rejecting samples with a high predictive variance, i.e. samples with high uncertainty. If the predictive variance estimates are well calibrated, as the variance threshold increases, fewer samples with high uncertainty are rejected and the accuracy should decrease.

(TFIDF) transformation. The prediction targets are the topics of the scientific papers in the networks. For the base kernel of the GP, we use a degree 3 polynomial kernel on the TFIDF features, which has been empirically shown to work well with similar models. The wavelet kernel uses two band pass-filters and a low-pass filter. The wavelet kernel is approximated with a degree 5 polynomial for Cora and Citeseer and with a degree 3 polynomial for PubMed. Moreover, for Cora and Citeseer, the kernel is used as part of a non-sparse variational GP, whereas for PubMed we use a sparse variational GP to enable stochastic optimisation of the ELBO using mini-batches.

The hyperparameters of the model are the initial band-pass scales and whether a low-pass filter should be included in the kernel. We train all GP models for up to 300 epochs with a learning rate of 0.01. To check convergence, we plot the ELBO curves in Figure 14 in the Appendix. Early-stopping and model selection are performed using the ELBO achieved on the training set and WGGP hence does not require a hold-out validation set. Similar to Ng et al. (2018), we thus also report the result of WGGP trained on both the training and the validation set and refer to it as WGGP-X. The results are presented in Table 1 where our model is very competitive against a set of state-of-the-art baselines including graph neural network and GP models. In particular, LanczosNet and AdaLanczosNet (Liao et al., 2019) were included as, like the method proposed here, they are designed to extract multi-scale information from graphs. We also included a version of our model called ChebGP, which uses Chebyshev polynomials for the spectrum approximation method, to show the superiority of the polynomial approximation method we adopted. Our model outperforms both a vanilla GP model operating solely on the node features and the Graph Gaussian process (GGP) (Ng et al., 2018) aggregating information from the first-hop neighbourhood, thus highlighting the benefit of our multi-scale approach. Additional results and ablation studies are presented in the Appendix.

Uncertainty Estimates. Unlike the neural network baselines, our proposed GP model performs approximate Bayesian inference and therefore outputs confidence estimates for its predictions at each node v_i via the variance of the variational predictive distribution $q(y_i)$. We expect reliable variance estimates to be useful in deciding which samples to reject (and potentially send to a human labeller) because the model is unable to make a prediction with high enough confidence. We evaluate our model in this regard by computing its predictive accuracies for different variance thresholds. For a lower threshold, more low-confidence samples are rejected, which should lead to a higher predictive performance. We confirm that this property holds for the confidence estimates of our model via Figure 7.

6 DISCUSSION

In integrating wavelets with a GP, we have developed a model that is capable of capturing multi-scale information in the data. By including different wavelet scales, the model combines various levels of localisation on graphs to capture beyond low-frequency elements. Even though the function is defined in the graph spectral domain, by adopting a polynomial approximation we avoid an expensive eigen-decomposition, allowing the model to scale to larger graphs. We show on synthetically generated data that different scales can be recovered accurately, and the multi-scale approach leads to competitive performance on real graph data sets against state-of-the-art graph models.

Applying the proposed wavelet model to a task at hand requires taking a number of practical considerations into account. Firstly, the number of scales in the wavelet kernel should ideally be chosen in a way such that the multi-scale graph data is captured by the different scales of the wavelets (although the model is robust to varying number of scales, cf. Appendix). For example, we may aim to match the number of scales in the kernel with the number of gaps in the spectrum by estimating the eigenvalue distribution of the graph Laplacian, which is already part of the wavelet transform approximation. Secondly, given the nature of wavelets as dilated and shifted band-pass filters, an interesting question is which mother wavelet to choose for the GP model. While our model is robust to different choices of the mother wavelets (cf. Appendix) certain options might be preferred for a given task based on their localisation properties in the spatial and spectral domain. Finally, which nodes are selected for training can impact the learning process and final performance. If domain knowledge is available, one may look to find strategic ways to sample training nodes that will lead to the best possible characterisation of input data given a limited sampling budget.

Acknowledgements

We would like to thank reviewers for their helpful comments. FLO gratefully acknowledges funding from the Huawei Hisilicon Studentship at the Department of Computer Science and Technology of the University of Cambridge. YCZ would like to thank the support from Oxford Networks & Society Group, Oxford-Man Institute, and EPSRC (EP/S515541/1). XD gratefully acknowledges support from the Oxford-Man Institute and the EPSRC (EP/T023333/1).

References

D. Bo, X. Wang, C. Shi, and H. Shen. Beyond low-frequency information in graph convolutional networks. In *AAAI Conference on Artificial Intelligence*, 2021.

V. Borovitskiy, I. Azangulov, A. Terenin, P. Mostowsky, M. P. Deisenroth, and N. Durrande. Matern Gaussian processes on graphs. In *International Conference on Artificial Intelligence and Statistics*, 2021.

M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković. *Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges*. MIT Press, 2021.

J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. Spectral networks and locally connected networks on graphs. In *International Conference on Learning Representations*, 2014.

P. Cheng, Y. Li, X. Zhang, L. Chen, D. Carlson, and L. Carin. Dynamic embedding on textual networks via a gaussian process. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7562–7569, 2020.

A. Clauset, C. Moore, and M. E. J. Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453:98–101, 2008.

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

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

E. Di Napoli, E. Polizzi, and Y. Saad. Efficient estimation of eigenvalue counts in an interval. *Numerical Linear Algebra with Applications*, 2016.

Y. Dong, K. Ding, B. Jalaian, S. Ji, and J. Li. Graph neural networks with adaptive frequency response filter. In *arXiv*, 2021.

J. Dutkowski, M. Kramer, M. A. Surma, R. Balakrishnan, J. M. Cherry, N. J. Krogan, and T. Ideker.

A gene ontology inferred from molecular networks. *Nature Biotechnology*, 31:38–45, 2013.

T. Fan, D. I. Shuman, S. Ubaru, and Y. Saad. Spectrum-adapted polynomial approximation for matrix functions with applications in graph signal processing. *Algorithms*, 13(11), 2020.

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

F. Gao, G. Wolf, and M. Hirn. Geometric scattering for graph data analysis. In *International Conference on Machine Learning*, 2019.

A. Girard. A fast 'monte-carlo cross-validation' procedure for large least squares problems with noisy data. *Numerical Mathematics*, 1989.

D. K. Hammond, P. Vandergheynst, and R. Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2): 129–150, 2011a.

D. K. Hammond, P. Vandergheynst, and R. Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, 2011b.

J. Hensman, A. G. de G. Matthews, and Z. Ghahramani. Scalable variational gaussian process classification. In *AISTATS*, 2015.

T. N. Kipf and M. Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*, 2017.

N. Li, W. Li, J. Sun, Y. Gao, Y. Jiang, and S.-T. Xia. Stochastic deep Gaussian processes over graphs. In *Advances in Neural Information Processing Systems 33*, 2020.

Q. Li, Z. Han, and X.-M. Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *AAAI Conference on Artificial Intelligence*, 2018.

S. Li, Y. Jin, and D. I. Shuman. Scalable M-Channel Critically Sampled Filter Banks for Graph Signals, 2019.

R. Liao, Z. Zhao, R. Urtasun, and R. Zemel. Lanczosnet: Multi-scale deep graph convolutional networks. In *International Conference on Learning Representations*, 2019.

L. Lin, Y. Saad, and C. Yang. Approximating spectral densities of large matrices. *SIAM Review*, 2016.

Z.-Y. Liu, S.-Y. Li, S. Chen, Y. Hu, and S.-J. Huang. Uncertainty aware graph gaussian process for semi-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4957–4964, 2020.

P. V. Mieghem. *Graph Spectra for Complex Networks*. Cambridge University Press, USA, 2011.

Y. Min, F. Wenkel, and G. Wolf. Scattering gcn: Overcoming oversmoothness in graph convolutional networks. In *Advances in Neural Information Processing Systems 33*, 2020.

H. Mingguo, W. Zhewei, H. Zengfeng, and X. Hongten. Bernnet: Learning arbitrary graph spectral filters via bernstein approximation. In *arXiv*, 2021.

Y. C. Ng, N. Colombo, and R. Silva. Bayesian semi-supervised learning with graph Gaussian processes. In *Advances in Neural Information Processing Systems 31*, 2018.

H. NT, T. Maehara, and T. Murata. Stacked graph filter. In *arXiv*, 2020.

K. Oono and T. Suzuki. Graph neural networks exponentially lose expressive power for node classification. In *International Conference on Learning Representations*, 2020.

F. L. Opolka and P. Liò. Graph convolutional Gaussian processes for link prediction. In *ICML Workshop on Graph Representation Learning and Beyond*, 2020.

G. Puy and P. Pérez. Structured sampling and fast reconstruction of smooth graph signals. *Information and Inference: A Journal of the IMA*, 2018.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

E. Ravasz and A.-L. Barabási. Hierarchical organization in complex networks. *Physical Review E*, 67(2): 026112, 2003.

P. Sen, G. M. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad. Collective classification in network data. *AI Magazine*, 2008.

D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst. 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, 2013.

D. I. Shuman, C. Wiesmeyr, N. Holighaus, and P. Vandergheynst. Spectrum-adapted tight graph wavelet and vertex-frequency frames. *IEEE Transactions on Signal Processing*, 2015.

R. N. Silver and H. Röder. Densities of States of Mega-Dimensional Hamiltonian Matrices. *International Journal of Modern Physics C*, 1994.

R. N. Silver, A. F. Voter, J. D. Kress, and H. Roeder. Kernel polynomial approximations for densities of states and spectral functions. 1996.

A. Smola and R. Kondor. Kernels and regularization on graphs. In *Annual Conference on Computational Learning Theory*, 2003.

N. Tremblay and P. Borgnat. Graph wavelets for multiscale community mining. *IEEE Transactions on Signal Processing*, 62(20):5227–5239, 2014.

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

A. Venkitaraman, S. Chatterjee, and P. Handel. Gaussian processes over graphs. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020.

I. Walker and B. Glocker. Graph convolutional Gaussian processes. In *International Conference on Machine Learning*, 2019.

L.-W. Wang. Calculating the density of states and optical-absorption spectra of large quantum systems by the plane-wave moments method. *Physical Review B*, 1994.

F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger. Simplifying graph convolutional networks. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

B. Xu, H. Shen, Q. Cao, Y. Qiu, and X. Cheng. Graph wavelet neural network. *ArXiv*, abs/1904.07785, 2019.

X. Zheng, B. Zhou, J. Gao, Y. G. Wang, P. Lio, M. Li, and G. Montúfar. How framelets enhance graph neural networks. *arXiv preprint arXiv:2102.06986*, 2021.

Y.-C. Zhi, Y. C. Ng, and X. Dong. Gaussian processes on graphs via spectral kernel learning. *arXiv:2006.07361*, 2020.

J. Zhu, Y. Yan, L. Zhao, M. Heimann, L. Akoglu, and D. Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. In *Advances in Neural Information Processing Systems 33*, 2020.

Supplementary Material: Adaptive Gaussian Processes on Graphs via Spectral Graph Wavelets

A VISUALISATION OF WAVELET TRANSFORM ON A REGULAR GRID

By applying the wavelet transforms to an impulse function centred around a certain node, we can visualise how wavelets of different scales spread around the centre node, capturing different ranges of neighbourhoods (cf. Figure 1). When applied to a regular grid graph, the pattern resembles that of the Euclidean domain. This is shown in Figure 8, where we apply the Mexican Hat wavelet transform with various scales to show the different ranges of neighbourhoods. Neighbours are weighted continuously with intensity becoming zero once beyond a certain proximity. Thus, by using different scales, we can capture different ranges of neighbourhood information.

B ADDITIONAL EXPERIMENTAL RESULTS

B.1 Synthetic Scale Recovery Experiments and Implementation Details

We run our synthetic experiments multiple times to show the overall behaviour of the model. We sample the labels for the nodes by Eq. (17) 100 times; for each sample, we also randomly select a set of nodes to use for training. The hyperparameters are optimised as part of the training process. For each set of training labels, we test 20 different initializations and use the converged values that lead to the lowest loss.

The selection of nodes for training will have an effect on the scales we recover. We have presented one particular random split for 50% of nodes used for training in Figure 4 of the main text; In Figures 9, 10 and 11, we present the scale recovery results for 10%, 30% and 70% of nodes selected for training for three random splits each. We can see the quality of the recoveries improves as the percentage of training nodes increases. Additionally, the approximate recoveries are consistently very close to the exact recoveries, showing the accuracy of our polynomial approximation.

B.2 Baseline GP Models on Synthetic Data

We also evaluate the baseline GP models from Section 5 on synthetic data. The graph neural network models were not compared against as they require a validation set of nodes, which are not assigned and would make an unfair comparison. We use GGP and ChebGP to make predictions on the synthetically generated signals, with the MAEs presented in Figure 12.

The results in Figure 12 show that the GGP only improves marginally with additional training data, indicating the model's inability to capture multi-scale information. ChebGP, which uses Chebyshev polynomials for approximations, does approximate a multi-scale spectral wavelet function, but we can see by the means and quantiles of the boxplots that they are less consistent in producing low MAEs compared to our polynomial approximation. As the number of training nodes increases, the model should be able to capture the different scales more accurately; however, the wider quantiles indicate the Chebyshev approximation is less consistent in producing accurate recoveries.

B.3 Performance on Synthetic Data Generated Using Different Ground Truth Wavelets

The synthetic setting described in Section 5 uses the Mexican Hat kernel in both the inference GP and the data generating model. We now study the case where there is a mismatch in mother wavelet between the inference GP and the data generating model. In Figure13 we always use a Mexican Hat wavelet for the inference GP and compare the case of using a Mexican Hat wavelet (Figures 13a and 13c) versus a Morlet wavelet (Figures

13b and 13d) in the data generating GP both in terms of prediction MAE (Figures 13a and 13b) and MAE of reconstructed filter compared to ground truth filter (Figures 13c and 13d).

B.4 WGGP without Feature Space Kernel

To measure the importance of the feature space kernel, we repeat experiments with WGGP on Cora and Citeseer with the feature space kernel $K_\Psi(\mathbf{x}_i, \mathbf{x}_j) = \delta_{ij}$ set to the identity. As a result, the model classifies nodes no longer also based on node features but on graph structure alone. We compare the results to those of the full WGGP model in Table 2. As expected, the performance of the model drops decisively when removing the dependence on the node features, demonstrating the importance of the node feature kernel for the predictive performance.

Method	**Cora**	**Citeseer**
WGGP	84.7	70.8
WGGP without node features	71.9	47.7

Table 2: Classification accuracy of the WGGP model with and without the node feature kernel. When removing the node feature kernel, the predictive performance drops by more than 10% for both data sets.

B.5 ELBO Plots

As described in Section 5, early stopping is performed based on the ELBO. To check convergence, we show how the ELBO varies from epoch to epoch in Figure 14. Note that the ELBO curve for the PubMed data set is non-monotonic as stochastic optimisation is employed during training.

B.6 Robustness Analysis

We perform a robustness analysis examining how the model performance changes as we vary different parts of the model or training setup, while keeping everything else as described in Section 5.

Data Split In a first experiment, we use 10 different data splits for Cora and Citeseer that retain the uniform distribution of classes and re-run the model with otherwise equal hyperparameters. The average performance across the 10 data splits is reported in Table 3 together with one standard deviation. We find that the model perofrmance only varies modestly for both data sets and the performance remains comparable to the one achieved on the public data split.

Number of scales We also analyse how the model performance varies when using different number of scales in the model, ranging from using only a low-pass filter to also including 4 band pass filters. The results are again reported in Table 3 with the standard deviation over the 4 different setups (0-4 scales), showing that the model accuracy varies only slightly when using different number of filters.

Hyperparameter initialisations Finally, we repeat the experiments with random initialisations of the scale hyperparameters. The results with their standard deviation over 10 different initialisations (Table 3) demonstrate the model's robustness to different hyperparameter initialisations.

Method	**Cora**	**Citeseer**
WGGP with varying data splits	82.4 ± 1.1	67.8 ± 2.7
WGGP with varying number of scales	84.7 ± 0.2	70.6 ± 0.2
WGGP with varying hyperparameter initalisations	84.2 ± 0.4	71.0 ± 0.6

Table 3: Results of the robustness analysis of the WGGP model when varying the data split, the number of scales, or the scale hyperparameter initialisations.

C DATA SET STATISTICS

Data	Type	$\mathbf{N_{nodes}}$	$\mathbf{N_{edges}}$	$\mathbf{N_{label\_cat}}$	$\mathbf{D_{features}}$	**Label Rate**
Cora	Citation	2,708	5,429	7	1,433	0.052
Citeseer	Citation	3,327	4,732	6	3,703	0.036
PubMed	Citation	19,717	44,338	3	500	0.003

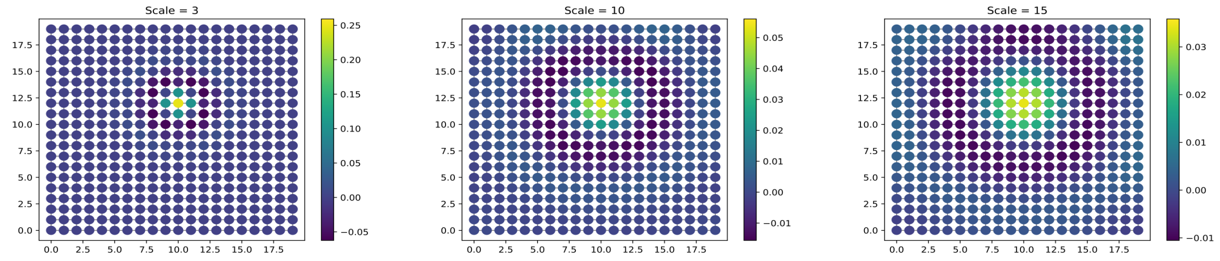
Table 4: Summary of citation networks for node classification experiments.

D COMPUTING PLATFORM AND CODE IMPLEMENTATION

The experiments were performed using Xeon W-2133 12GB NVIDIA GTX 1080 Ti and 48GB NVIDIA Quadro RTX 8000.

The code for reproducing the results in the paper has been submitted as part of the supplementary material.

E FIGURES



(a) (b) (c)

Figure 8: The Mexican Hat wavelet transform of a δ signal on a regular grid graph. The grid simulates a Euclidean domain to demonstrate the neighbourhoods more clearly at different scales.

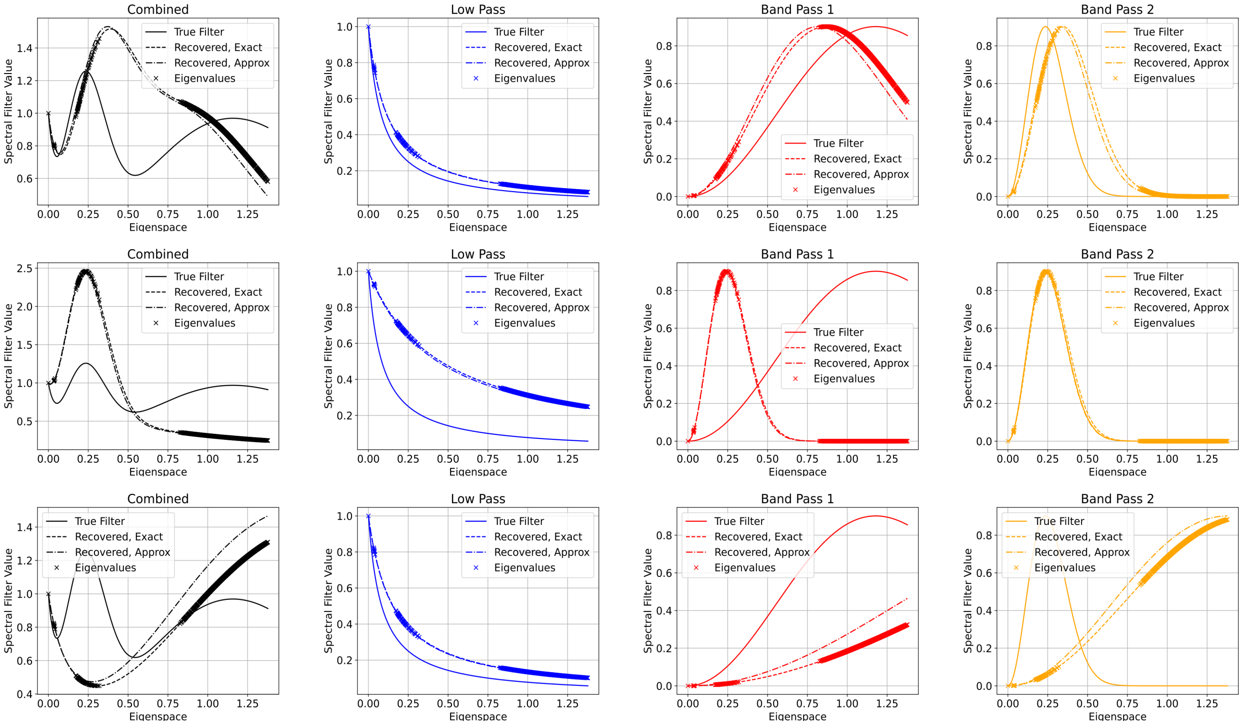


Figure 9: Scale recoveries from synthetic experiments using 10% of nodes as training. Each row is a different random selections of training nodes.

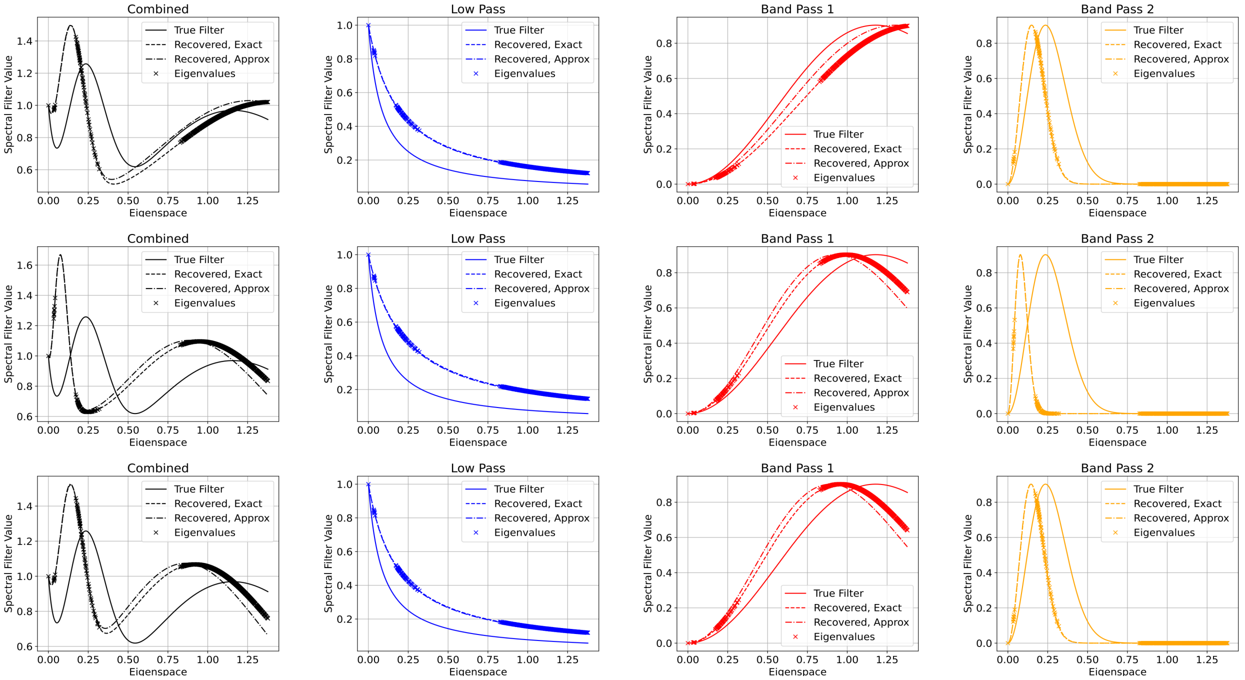


Figure 10: Scale recoveries from synthetic experiments using 30% of nodes as training. Each row is a different random selections of training nodes.

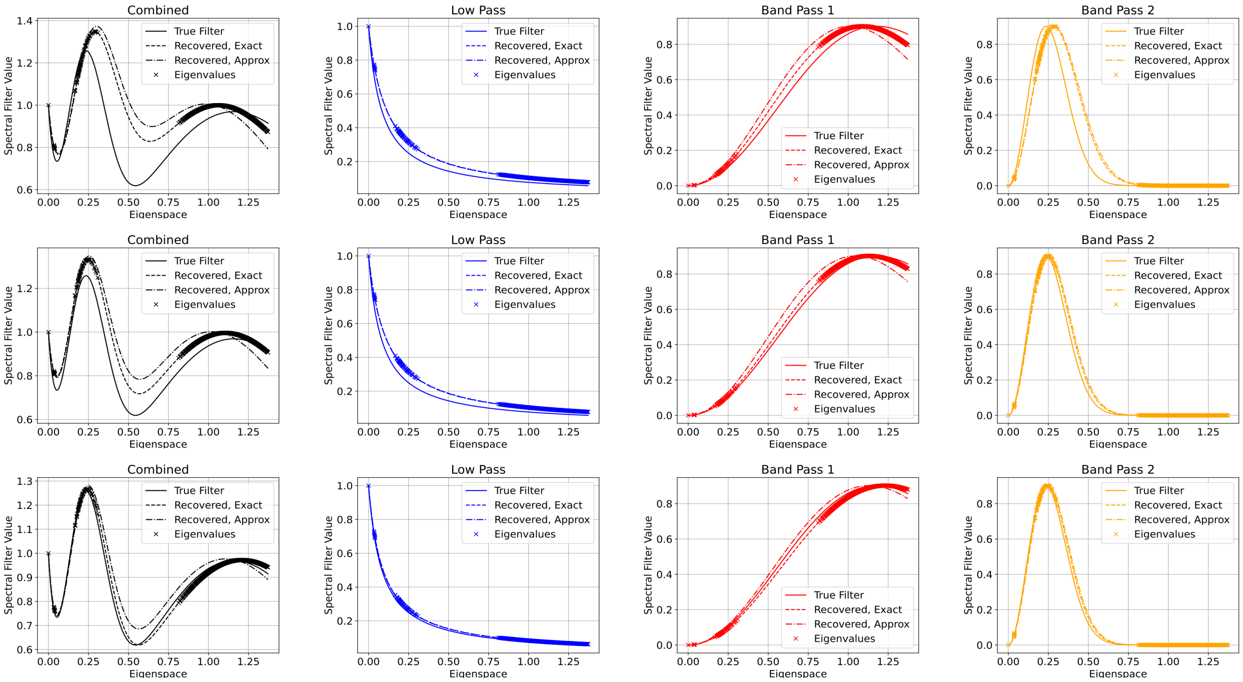
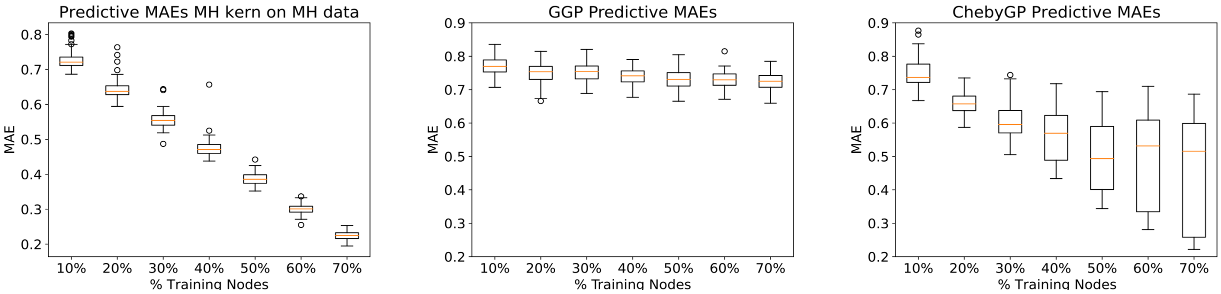
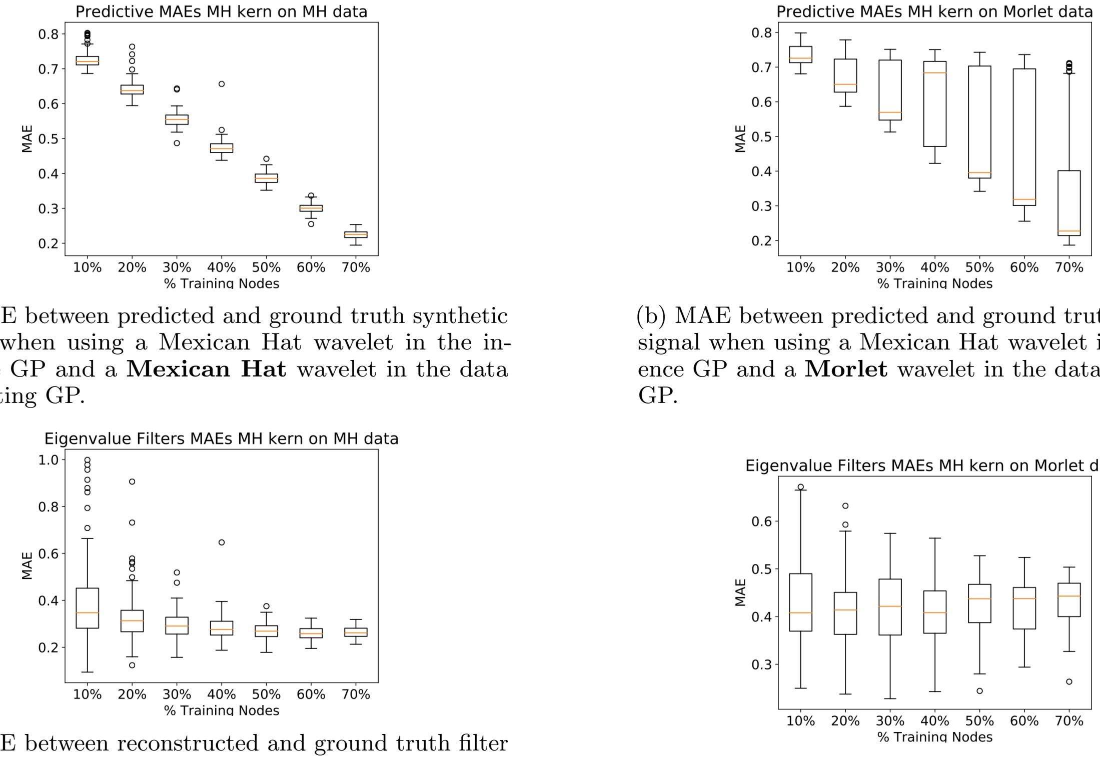


Figure 11: Scale recoveries from synthetic experiments using 70% of nodes as training. Each row is a different random selections of training nodes.



(a) WGGP (b) GGP (c) ChebGP

Figure 12: WGGP prediction MAE on synthetic data (a and identical to Figure 3b in main text) compared to MAEs of baseline GP models, GGP (b) and ChebGP (c).



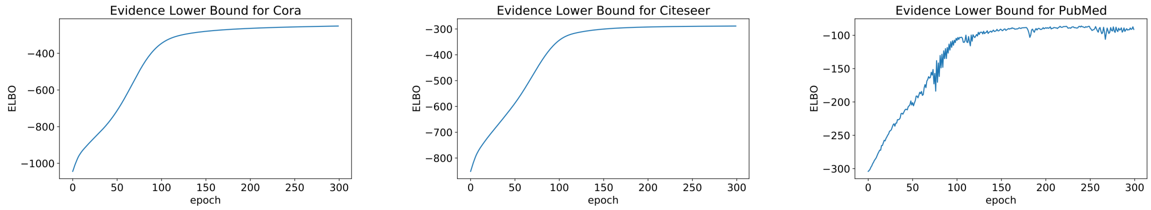
(a) MAE between predicted and ground truth synthetic signal when using a Mexican Hat wavelet in the inference GP and a **Mexican Hat** wavelet in the data generating GP.

(b) MAE between predicted and ground truth synthetic signal when using a Mexican Hat wavelet in the inference GP and a **Morlet** wavelet in the data generating GP.

(c) MAE between reconstructed and ground truth filter when using a Mexican Hat wavelet in the inference GP and a **Mexican Hat** wavelet in the data generating GP.

(d) MAE between reconstructed and ground truth filter when using a Mexican Hat wavelet in the inference GP and a **Morlet** wavelet in the data generating GP.

Figure 13: Comparing the prediction and filter MAEs for different fractions of nodes used for trainig when the choice of mother wavelet match or do not match between the inference GP and the data generating GP.



(a) Cora

(b) Citeseer

(c) Pubmed

Figure 14: Value of the ELBO during training over time.