Matérn Gaussian Processes on Graphs

Viacheslav Borovitskiy[*1, 5] Iskander Azangulov[*1] Alexander Terenin[*2]
Peter Mostowsky[1] Marc Peter Deisenroth[3] Nicolas Durrande[4]

[1]St. Petersburg State University [2]Imperial College London [3]University College London [4]Secondmind
[5]St. Petersburg Department of Steklov Mathematical Institute of Russian Academy of Sciences

arXiv:2010.15538v1 [stat.ML] 29 Oct 2020

Abstract

Gaussian processes are a versatile framework for learning unknown functions in a manner that permits one to utilize prior information about their properties. Although many different Gaussian process models are readily available when the input space is Euclidean, the choice is much more limited for Gaussian processes whose input space is an undirected graph. In this work, we leverage the stochastic partial differential equation characterization of Matérn Gaussian processes—a widely-used model class in the Euclidean setting—to study their analog for undirected graphs. We show that the resulting Gaussian processes inherit various attractive properties of their Euclidean and Riemannian analogs and provide techniques that allow them to be trained using standard methods, such as inducing points. This enables graph Matérn Gaussian processes to be employed in mini-batch and non-conjugate settings, thereby making them more accessible to practitioners and easier to deploy within larger learning frameworks.

1 Introduction

Gaussian process (GP) models have become ubiquitous in machine learning, and have been shown to be a data efficient approach in a wide variety of applications (Rasmussen and Williams, 2006). Key elements behind the success of GP models include their ability to assess and propagate uncertainty, as well as encode different kinds of prior information about the function they seek to approximate. For example, by choosing different covariance kernels, one can encode different degrees of differentiability, or specific patterns, such as periodicity and symmetry. Although the input and output spaces of GP models are typically subsets of $\mathbb{R}$ or $\mathbb{R}^d$, this is by no means a restriction of the GP framework, and it is possible to define models for other types of input or output spaces (Lindgren et al., 2011; Mallasto and Feragen, 2018; Borovitskiy et al., 2020).

In many applications, such as predicting street congestion within a road network, using kernels based on the Euclidean distance between two locations in the city does not make much sense. In particular, locations that are spatially close may have different traffic patterns, for instance, if two nearby roads are disconnected or if traffic is present only in one direction of travel. Here, it is much more natural for the model to account directly for a distance based on the graph structure. In this work, we study GPs whose inputs or outputs are indexed by the vertices of an undirected graph, where each edge between adjacent nodes is assigned a positive weight.

Gaussian Markov random fields (GMRF) (Rue and Held, 2005) provide a canonical framework for such settings. A GMRF builds a graph GP by introducing a Markov structure on the graph's vertices, and results in models that are computationally efficient. These constructions are well-defined and effective, but they require Markovian assumptions which limit model flexibility. Although it may be tempting to replace the Euclidean distance that can typically be found in the expression of stationary kernels by the graph distance, this typically does not result in a well-defined covariance kernel (Feragen et al., 2015).

As a consequence, working with GPs on graphs requires one to define bespoke kernels, which for graphs with finite sets of nodes can be viewed as parameterized structured covariance matrices that encode dependence between vertices. A few covariance structures dedicated to graph GPs have been explored in the lit-

[*]Equal contribution. Preprint. Copyright by the author(s).

erature, such as the diffusion kernel or random walk kernels (Kondor and Lafferty, 2002; Vishwanathan et al., 2010), but the available choices are limited compared to typical Euclidean input spaces and this results in impaired modeling abilities.

In this work, we study graph analogs of kernels from the Matérn family, which are among the most commonly used kernels for Euclidean input spaces (Rasmussen and Williams, 2006; Stein, 1999). These can be used as GP input covariances, or GP output cross-covariances. Our approach is similar to Whittle (1963), Lindgren et al. (2011), and Borovitskiy et al. (2020), where GPs with Matérn kernels are defined on Euclidean and Riemannian manifolds via their stochastic partial differential equation (SPDE) representation. For the graph Matérn GPs with integer smoothness parameters we obtain sparse precision matrices that can be exploited for improving the computational speed. For example, they can benefit from the well-established GMRF framework (Rue and Held, 2005) or from recent advances in non-conjugate GP inference on graphs (Durrande et al., 2019). As an alternative, we present a Fourier feature approach to building Matérn kernels on graph with its own set of advantages, such as hyperparameter optimization without incurring the cost of computing a matrix inverse at each optimization step. We also discuss important properties of the graph Matérn kernels, such as their convergence to the Euclidean and Riemannian Matérn kernels when the graph becomes more and more dense.

2 Gaussian processes

Let X be a set. A random function $f : X \to \mathbb{R}$ is a Gaussian process $f \sim \mathrm{GP}(\mu, k)$ with mean function $\mu(\cdot)$ and kernel $k(\cdot, \cdot)$ if, for any finite set of points $\boldsymbol{x} \in X^n$, the random vector $f(\boldsymbol{x})$ is multivariate Gaussian with mean vector $\boldsymbol{\mu} = \mu(\boldsymbol{x})$ and covariance matrix $\mathbf{K}_{\boldsymbol{xx}} = k(\boldsymbol{x}, \boldsymbol{x})$. Without loss of generality, we assume that the prior mean μ is zero.

For a given set of training data (x_i, y_i), we define the model $y_i = f(x_i) + \varepsilon_i$ where $f \sim \mathrm{GP}(0, k)$ and $\varepsilon_i \sim \mathrm{N}(0, \sigma^2)$. The posterior of f given the observations is another GP. Its conditional mean and covariance are

$$\mu_{|\boldsymbol{y}}(\cdot) = \mathbf{K}_{\cdot\boldsymbol{x}}(\mathbf{K}_{\boldsymbol{xx}} + \sigma^2\mathbf{I})^{-1}\boldsymbol{y} \tag{1}$$

$$k_{|\boldsymbol{y}}(\cdot, \cdot') = k(\cdot, \cdot') - \mathbf{K}_{\cdot\boldsymbol{x}}(\mathbf{K}_{\boldsymbol{xx}} + \sigma^2\mathbf{I})^{-1}\mathbf{K}_{\boldsymbol{x}\cdot'} \tag{2}$$

which uniquely characterize the posterior distribution (Rasmussen and Williams, 2006). Following Wilson et al. (2020), posterior sample paths can be written as

$$f(\cdot) \mid \boldsymbol{y} = f(\cdot) + \mathbf{K}_{\cdot\boldsymbol{x}}(\mathbf{K}_{\boldsymbol{xx}} + \sigma^2\mathbf{I})^{-1}(\boldsymbol{y} - f(\boldsymbol{x}) - \boldsymbol{\varepsilon}) \tag{3}$$

where $(\cdot)$ is an arbitrary set of locations.

2.1 The Matérn kernel

When $X = \mathbb{R}^d$ and $\tau = x - x'$, Matérn kernels are defined as

$$k_\nu(x, x') = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu}\frac{\|\tau\|}{\kappa}\right)^\nu K_\nu\left(\sqrt{2\nu}\frac{\|\tau\|}{\kappa}\right) \tag{4}$$

where K_ν is the modified Bessel function of the second kind (Gradshteyn and Ryzhik, 2014). The parameters σ^2, κ and ν are positive scalars that have a natural interpretation: σ^2 is the variance of the GP, the length-scale κ controls how distances are measured in the input space, and ν determines mean-square differentiability of the GP (Rasmussen and Williams, 2006). As $\nu \to \infty$, the Matérn kernel converges to the widely-used squared exponential kernel

$$k_\infty(x, x') = \sigma^2 \exp\left(-\frac{\|\tau\|^2}{2\kappa^2}\right). \tag{5}$$

In our setting, an important property of Matérn kernels is their connection to stochastic partial differential equations (SPDEs). Whittle (1963) has shown that Matérn GPs on $X = \mathbb{R}^d$ satisfies the SPDE

$$\left(\frac{2\nu}{\kappa^2} - \Delta\right)^{\frac{\nu}{2} + \frac{d}{4}} f = \mathcal{W} \tag{6}$$

for $\nu < \infty$, where Δ is the Laplacian and $\mathcal{W}$ is Gaussian white noise (Lifshits, 2012) re-normalized by a certain constant—see Lindgren et al. (2011) or Borovitskiy et al. (2020) for details. Similarly, the limiting squared exponential GP satisfies

$$e^{-\frac{\kappa^2}{4}\Delta} f = \mathcal{W} \tag{7}$$

where $e^{-\frac{\kappa^2}{4}\Delta}$ is the (rescaled) heat semigroup (Evans, 2010; Grigoryan, 2009). These equations have been studied as a means to extend Matérn Gaussian processes to Riemannian manifolds such as the sphere and torus. Since these spaces can be discretized to form a graph, they play an important role in the sequel.

2.2 Gaussian processes on graphs

A number of approaches have been proposed to define GPs over a weighted undirected graph $G = (V, E)$. The aim is to define a GP indexed by the vertices V, which reflects the notion of *closeness* induced by the edges E and their associated weights. This has been studied by a variety of authors, including Kondor and Lafferty (2002) and Rue and Held (2005), and others, in areas such as GMRFs and diffusion kernels. A number of variations, such as introducing dependence on attributes contained in nodes (Ng et al., 2018), are possible. We will interpret the ideas presented in this work from multiple viewpoints in order to synthesize these different perspectives.

A number of authors, such as Venkitaraman et al. (2020) and Zhi et al. (2020) have studied graph-structured multi-output Gaussian process $f : \mathbb{R} \to \mathbb{R}^{|V|}$, where $|V|$ is the number of nodes for which the outputs dependencies should reflect the graph structure. Although such settings may appear different from the scope outlined above, the problem can be cast into the proposed graph GP framework by constructing a Gaussian process $f : \mathbb{R} \times V \to \mathbb{R}$ through partial function application.

It is also possible to consider Gaussian processes $f : \mathcal{G} \to \mathbb{R}$ where $\mathcal{G}$ is an appropriately defined *space of graphs*. Here, each individual input to the GP is an *entire graph* rather than just a node on a fixed graph. This setting departs significantly from the preceding ones, and we do not study it in this work—a recent survey on the topic is given by Kriege et al. (2020).

3 Matérn GPs on graphs

We now define the *Matérn* family of Gaussian processes on graphs by generalizing their SPDE characterization to the graph setting. This will entail introducing appropriate notions for the left-hand-side and right-hand-side of equations (6) and (7). Note that since a graph is a finite set, such a Gaussian process can be viewed as a multivariate Gaussian whose indices are the graph's nodes. Throughout this work, we refer to these as Gaussian processes to emphasize that their covariance reflects the structure of the graph.

Let G be a weighted undirected graph whose weights are non-negative—for an unweighted graph, assume all weights are equal to one. Denote its adjacency matrix by $\mathbf{W}$, its diagonal degree matrix by $\mathbf{D}$ with $\mathbf{D}_{ii} = \sum_j W_{ij}$, and define the *graph Laplacian* by

$$\boldsymbol{\Delta} = \mathbf{D} - \mathbf{W}. \tag{8}$$

The graph Laplacian is a symmetric, positive semi-definite matrix, which we view as a linear operator acting on a $|V|$-dimensional real space. Note that this operator should be viewed as an analog of $-\Delta$ in the Euclidean or Riemannian setting.[1] Notwithstanding the different sign convention, the graph and Euclidean Laplacian are intimately linked. For example, the Laplacian of a graph given by a grid corresponds to the finite-difference approximation of the Euclidean Laplacian, and diffusions on graphs involve the graph Laplacian in the same way diffusion in continuous spaces involve the classical Laplacian. If G is connected, then $\boldsymbol{\Delta}$ has a nullspace of dimension 1, following the case of compact manifolds (Smola and Kondor, 2003). Since $\boldsymbol{\Delta}$ is symmetric positive semi-definite, it admits an eigenvalue decomposition $\boldsymbol{\Delta} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$ where $\boldsymbol{\Lambda}$ is diagonal with non-negative entries and $\mathbf{U}$ is orthogonal.

To construct graph-theoretic analogs of the differential operators in (6) and (7), we introduce a notion of *functional calculus* for $\boldsymbol{\Delta}$. Let $\Phi : \mathbb{R} \to \mathbb{R}$ be a function. For the diagonal matrix $\boldsymbol{\Lambda}$, let $\Phi(\boldsymbol{\Lambda})$ be a diagonal matrix defined by applying Φ to the diagonal of Λ element-wise. Define the matrix $\Phi(\boldsymbol{\Delta})$ to be

$$\Phi(\boldsymbol{\Delta}) = \mathbf{U}\Phi(\boldsymbol{\Lambda})\mathbf{U}^T. \tag{9}$$

Although such a definition may seem arbitrary, this generalization of Φ to square matrices can be interpreted intuitively as plugging $\boldsymbol{\Delta}$ into the Taylor expansion of Φ, which immediately boils down to (9) due to the orthogonality of $\mathbf{U}$. Taking Φ to be one of

$$\Phi(\lambda) = \left(\frac{2\nu}{\kappa^2} + \lambda\right)^{\frac{\nu}{2}} \qquad \Phi(\lambda) = e^{\frac{\kappa^2}{4}\lambda} \tag{10}$$

gives the operators on the left-hand side of (6) and (7), respectively.[2] Note that the term $d/4$ present in the Euclidean and Riemannian cases to ensure regularity is not needed here. This will result in a slightly different scaling for graph Matérn kernels, which for Markovian cases are indexed by integers rather than the half-integers.

Replacing Gaussian white noise process with a standard Gaussian $\boldsymbol{\mathcal{W}} \sim \mathrm{N}(\mathbf{0}, \mathbf{I})$ gives the equations

$$\left(\frac{2\nu}{\kappa^2} + \boldsymbol{\Delta}\right)^{\frac{\nu}{2}} \boldsymbol{f} = \boldsymbol{\mathcal{W}}, \qquad e^{\frac{\kappa^2}{4}\boldsymbol{\Delta}} \boldsymbol{f} = \boldsymbol{\mathcal{W}}, \tag{11}$$

where we associate the vector of coefficients $\boldsymbol{f}$ with the function $f : V \to \mathbb{R}$, which gives the graph Gaussian process of interest. This gives

$$\boldsymbol{f} \sim \mathrm{N}\Big(\mathbf{0}, \left(\tfrac{2\nu}{\kappa^2} + \boldsymbol{\Delta}\right)^{-\nu}\Big), \quad \boldsymbol{f} \sim \mathrm{N}\Big(\mathbf{0}, e^{-\frac{\kappa^2}{2}\boldsymbol{\Delta}}\Big) \tag{12}$$

as *graph Matérn* and *graph squared exponential* Gaussian processes, respectively. Note that the former covariance is *not* obtained by adding $\frac{2\nu}{\kappa^2}$ to the entries of the matrix $\boldsymbol{\Delta}$, but rather by adding this number to its *eigenvalues*, in the sense of Equation (9). By construction their covariance matrices are positive semi-definite, and we refer to them as the *graph Matérn* and *graph diffusion* kernels, respectively. These possess a number of key properties, which we enumerate below.

[1] This sign convention ensures the positive semi-definiteness of $\boldsymbol{\Delta}$ and is equivalent to the analyst's (rather than geometer's) convention for studying the continuous Laplacian.

[2] We use $\Phi(\lambda) = \left(\frac{2\nu}{\kappa^2} + \lambda\right)^{\frac{\nu}{2}}$ and $\Phi(\lambda) = e^{\frac{\kappa^2}{4}\lambda}$ instead of $\Phi(\lambda) = \left(\frac{2\nu}{\kappa^2} - \lambda\right)^{\frac{\nu}{2}}$ and $\Phi(\lambda) = e^{-\frac{\kappa^2}{4}\lambda}$ because of the different sign convention for the graph Laplacian.

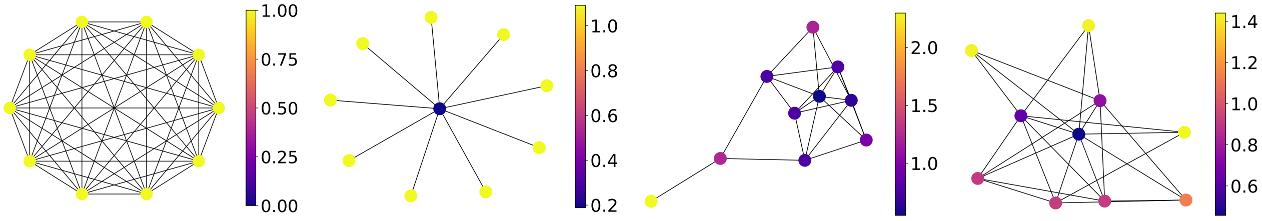


Figure 1: Here we illustrate prior variance for a complete graph, a star graph, and two randomly generated graphs. For the complete graph, the variance is uniform because of its symmetry (reordering the nodes always gives the same graph). In the star graph, the center node has much lower variance, this mirrors the behavior of random walk for which the return time is lower for the center node than for the other nodes. The random graphs illustrate the idea that variance depends on graph structure rather than on the degrees of individual nodes.

Sparsity. For sparse graphs, $\boldsymbol{\Delta}$ is sparse. Hence, for sufficiently small integers ν and most graphs, the precision matrices $\left(\frac{2\nu}{\kappa^2} + \boldsymbol{\Delta}\right)^{\nu}$ are sparse. This can be exploited to significantly accelerate their computational efficiency, and is utilized extensively by prior work on GMRFs (Rue and Held, 2005).

Non-uniform variance. The prior variance for the introduced kernels varies along vertices. This behavior is similar to the variance of the Matérn kernel on certain non-homogeneous Riemannian manifolds (Borovitskiy et al., 2020). Figure 1 illustrates the prior variance of the graph Matérn kernel on a set of graphs. It can be seen that the kernel's variance is not simply a function of degree and depends in a complex manner on the graph in question.

Urry and Sollich (2013) study a similar phenomenon in the context of *random walk kernels.* They show that the variance is determined by the return time of a certain random walk defined on the graph.

Use with symmetric normalized graph Laplacian. The above kernels are defined in terms of the graph Laplacian $\boldsymbol{\Delta}$. In some applications, it might be preferable to instead work with the *symmetric normalized graph Laplacian* $\mathbf{D}^{-1/2}\boldsymbol{\Delta}\mathbf{D}^{-1/2}$. Doing so in above expressions yields

$$\boldsymbol{f} \sim \mathrm{N}\left(\mathbf{0}, \left(\tfrac{2\nu}{\kappa^2} + \mathbf{D}^{-1/2}\boldsymbol{\Delta}\mathbf{D}^{-1/2}\right)^{-\nu}\right) \tag{13}$$

$$\boldsymbol{f} \sim \mathrm{N}\left(\mathbf{0}, e^{-\frac{\kappa^2}{2}\mathbf{D}^{-1/2}\boldsymbol{\Delta}\mathbf{D}^{-1/2}}\right) \tag{14}$$

which we call the *symmetric normalized graph Matérn GP* and the *symmetric normalized graph squared exponential GP*, respectively.

Connection with the graph diffusion equation. The graph diffusion kernel $e^{-\frac{\kappa^2}{2}\boldsymbol{\Delta}}$ is the Green's function of the graph diffusion equation. That is, if $\boldsymbol{\phi} : [0, \infty) \times V \to \mathbb{R}$ solves the differential equation

$$\frac{\mathrm{d}\boldsymbol{\phi}}{\mathrm{d}t} + \boldsymbol{\Delta}\boldsymbol{\phi} = 0 \qquad\qquad \boldsymbol{\phi}|_{t=0} = \boldsymbol{v} \tag{15}$$

then $\boldsymbol{\phi}|_{t=\tau} = e^{-\tau\boldsymbol{\Delta}}\boldsymbol{v}$. This equation describes heat transfer along the graph. If $\boldsymbol{\Delta}$ is replaced with the symmetric normalized graph Laplacian $D^{-1/2}\boldsymbol{\Delta}D^{-1/2}$, then the value $\boldsymbol{\phi}|_{t=\tau}$ can be interpreted as the unnormalized density of a continuous-time random walk which moves along the graph.

Limits and connection with random walks. We detail two different ways in which the graph diffusion kernel arises as a limit of a sequence of kernels.

First, mirroring the Euclidean case, the graph Matérn kernel converges to the graph diffusion kernel as $\nu \to \infty$, when scaled appropriately. To see this, note that the eigenvectors of both kernels are identical, and consider the eigenvalues

$$(2\nu/\kappa^2)^{\nu}\left(\frac{2\nu}{\kappa^2} + \lambda\right)^{-\nu} \xrightarrow{\nu\to\infty} e^{-\frac{\kappa^2}{2}\lambda}. \tag{16}$$

This implies that the corresponding matrices converge, and hence that the Gaussian processes converge in distribution.

Second, the graph diffusion kernel arises as a limit of the *random walk kernel* of Smola and Kondor (2003). This kernel is defined as

$$\left(\mathbf{I} - (1-\alpha)\mathbf{D}^{-1/2}\boldsymbol{\Delta}\mathbf{D}^{-1/2}\right)^{p}, \tag{17}$$

which arises by symmetrizing the p-step transition matrix of a lazy random walk on the graph. This kernel looks superficially similar to the Matérn kernel defined previously, but contains a number of important differences. In particular, the power p is positive rather than negative, the laziness parameter α is restricted to be in $[0, 1)$, and the Laplacian is subtracted rather than added. Nonetheless, this kernel also converges to the

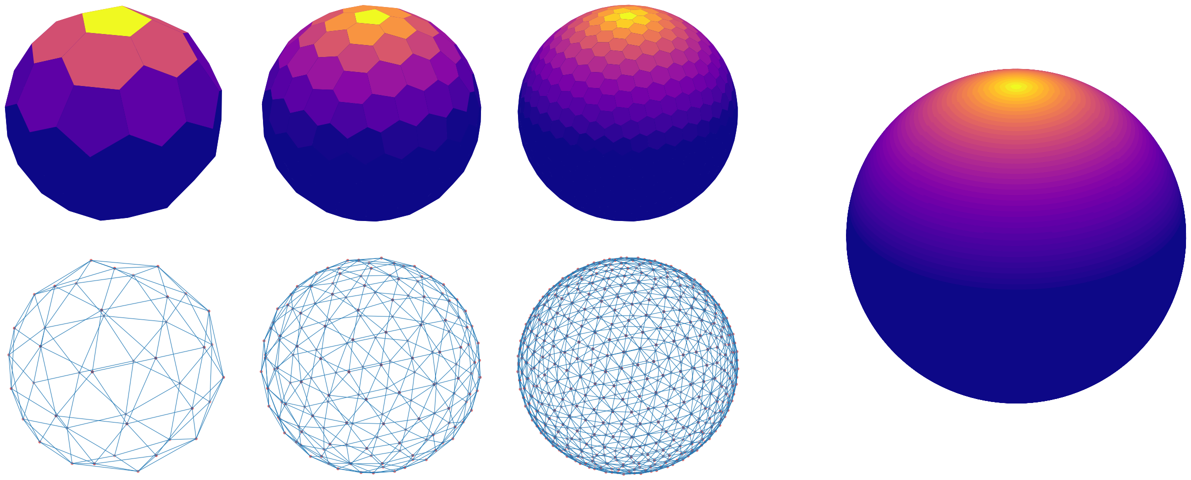


Figure 2: The Matérn kernel $k_{3/2}(x, \cdot)$ defined on increasingly fine triangular meshes (three left columns), and Matérn kernel $k_{1/2}(x, \cdot)$ on the sphere (rightmost). The true kernel is computed using 512 analytically known Laplace-Beltrami eigenpairs via the technique in Borovitskiy et al. (2020). The discrepancy in the smoothness parameter is due to a different parameterization of the kernel, which accounts for dimensionality of the Riemannian manifold. The point x is chosen to correspond to the north pole. For graphs defined by the meshes, kernel values are projected onto the sphere via piecewise-constant interpolation based on spherical Voronoi diagram induced by the graph nodes. The graph kernel converges to the true kernel on the right.

diffusion kernel, in the limit $p \to \infty$ with $\alpha = 1 - \frac{\kappa^2}{2p}$ where κ is a fixed constant. Specifically,

$$\begin{aligned}&\left(\mathbf{I} - (1-\alpha)\mathbf{D}^{-1/2}\mathbf{\Delta}\mathbf{D}^{-1/2}\right)^p \\ &\qquad \xrightarrow{p\to\infty} e^{-\frac{\kappa^2}{2}\mathbf{D}^{-1/2}\mathbf{\Delta}\mathbf{D}^{-1/2}}\end{aligned} \tag{18}$$

which yields the normalized graph diffusion kernel.

Relationship with Matérn Gaussian processes on Riemannian manifolds. One of the most common ways that linear (S)PDEs in the Euclidean and Riemannian settings are solved is via the *finite element method*. Roughly speaking, this technique expresses the (S)PDE solution using a truncated basis expansion, whose coefficients are obtained by solving a sparse linear system. A typical choice for the finite element basis are *piecewise polynomial* functions constructed using the nodes and edges of a graph, which corresponds to a mesh discretization of the manifold.

By their construction, the basis weights of a finite-element-discretized Riemannian Matérn Gaussian process define a GMRF approximation to the Matérn Gaussian process. The precise form of the GMRF approximation obtained depends on the choice of finite element space. Under appropriate regularity conditions, Lindgren et al. (2011) have proven convergence of the finite element discretizations to the solution of the true SPDE solution.

Not all kernels that arise this way are graph Matérn kernels. Fortunately, a number of other convergence results (Belkin and Niyogi, 2007; Burago et al., 2013) are available. One can show under appropriate regularity conditions and proper weighting of edges that the eigenvalues and eigenvectors of the graph Laplacian converge to the eigenvalues and eigenfunctions of the Laplace–Beltrami operator. Using these ideas, Sanz-Alonso and Yang (2020) provide a theoretical framework for studying how graph Matérn kernels converge to their Riemannian counterparts. This is illustrated in Figure 2, which shows convergence of a graph Matérn kernel to its Riemannian limit for a sphere.

Use as an output cross-covariance. The above section describes Matérn Gaussian processes to model functions $f : V \to \mathbb{R}$. We now briefly recall how to lift this construction to functions $\boldsymbol{f} : \mathbb{R}^d \to \mathbb{R}^{|V|}$, where the output cross-covariance is induced by the graph G. The simplest way to do so is to apply the *intrinsic coregionalization model* (Alvarez et al., 2011). Here, one views a multi-output function $\boldsymbol{f}(x)$ as a single-output function with an additional input $f(x, i)$ such that $\boldsymbol{f}(x)_i = f(x, i)$. One then uses a separable kernel $k((x, i), (x', i')) = k_{\mathbb{R}^d}(x, x')k_G(i, i')$, where $k_G(i, i')$ is the graph kernel. This defines model that respects the graph structure over the output space.

3.1 Scalable Training

Here we discuss how to train Matérn Gaussian processes in a scalable manner. The techniques we present fall in one of two broad areas: (1) graph Fourier feature approximations, and (2) efficient computation via sparse linear algebra. All of the techniques described here apply even if the graph is too large to store its full covariance matrix in memory.

Graph Fourier feature methods approximate the kernel matrix using a truncated eigenvalue expansion. Here, we first obtain the ℓ smallest eigenvalues and eigenvectors of $\mathbf{\Delta}$ using a routine designed for efficiently computing the eigenvalues of sparse matrices, such as the Lanczos algorithm. Since the mappings

$$\lambda \mapsto \left(\frac{2\nu}{\kappa^2} + \lambda\right)^{-\nu} \qquad \lambda \mapsto e^{-\frac{\kappa^2}{4}\lambda} \tag{19}$$

are decreasing functions on $\mathbb{R}^+$, we thus obtain the ℓ largest eigenvalues of the Matérn or diffusion kernels. Once this is obtained, scalable training proceeds by relating the approximate GP with a Bayesian linear regression model, mirroring Euclidean Fourier features (Rahimi and Recht, 2008). The main difficulty with this approach is that it can exhibit *variance starvation* issues, which causes approximation quality to deteriorate in the large-data regime (Wang et al., 2018).

A different way to train graph GPs scalably is by restricting ν to small integer values, for which in most graphs the inverse kernel matrices, termed *precision* matrices

$$\left(\frac{2\nu}{\kappa^2} + \mathbf{\Delta}\right)^{\nu} \tag{20}$$

are sparse. The prior is then a *Gaussian–Markov random field* for which a variety of scalable training procedures are already available off-the-shelf (Rue and Held, 2005). This approach generally avoids variance starvation, at a cost often no worse than that of graph Fourier features. The main limitation is that the computational gains diminish when ν increases, and that the numerical routines needed for scalable training may not be well-supported by standard Gaussian process packages. This support is rapidly improving: as shown by Durrande et al. (2019) and Adam et al. (2020), it is possible to leverage certain sparse computations in automatic differentiation frameworks, such as TensorFlow to accelerate variational GP models and their sparse counterparts.

Non-conjugate learning via doubly stochastic variational inference. In non-conjugate settings, such as classification, the posterior distribution is no longer Gaussian, and the training techniques from the preceding sections do not apply. Here we briefly sketch how to recover analogous techniques using variational inference and inducing points.

Let $\boldsymbol{z} \subset V$ be a set of variational *inducing inputs* (Hensman, Fusi, et al., 2013), let $\boldsymbol{x} \subset V$ be the training locations, let $\boldsymbol{u} = \boldsymbol{f}(\boldsymbol{z})$. We approximate the distribution $p(\boldsymbol{u}, f \mid \boldsymbol{y})$ with a variational family $q(\boldsymbol{u}, f)$ of the form $q(\boldsymbol{u}, f) = p(f \mid \boldsymbol{u})q(\boldsymbol{u})$, and train the model by minimizing the Kullback–Leibler divergence

$$D_{\mathrm{KL}}(q(\boldsymbol{u}, f) \mid\mid p(\boldsymbol{u}, f \mid \boldsymbol{y})) = \mathop{\mathbb{E}}_{q(\boldsymbol{u},f)} \ln \frac{q(\boldsymbol{u}, f)}{p(\boldsymbol{u}, f \mid \boldsymbol{y})} \tag{21}$$

$$= D_{\mathrm{KL}}(q(\boldsymbol{u}) \mid\mid p(\boldsymbol{u})) - \sum_{i=1}^{N} \mathop{\mathbb{E}}_{q(\boldsymbol{u},f)} \ln p(y_i \mid f_i) \tag{22}$$

$$- \ln p(\boldsymbol{y})$$

where we henceforth ignore the term $\ln p(\boldsymbol{y})$ because it does not depend on q.

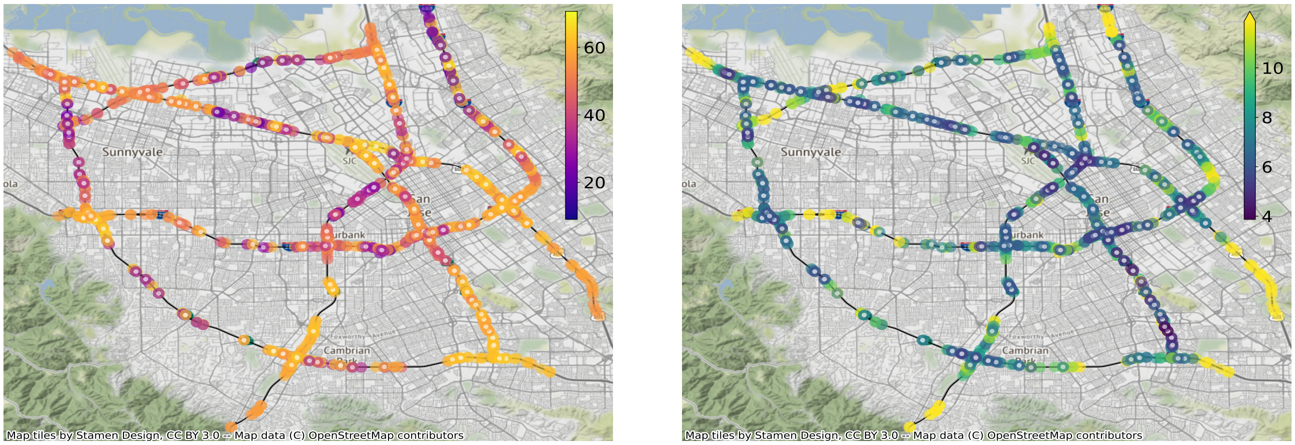
Following Titsias (2009), we choose $q(\boldsymbol{u}) \sim \mathrm{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ to be a free-form multivariate Gaussian. Minimizing the Kullback–Leibler divergence therefore entails finding the closest *GP posterior* to the true (non-Gaussian) posterior. The inducing points $\boldsymbol{z}$ can be chosen to be a subset of the data, or even the entire graph, provided one works with sparse precision matrices.

To minimize the Kullback–Leibler divergence, one employs a stochastic optimization algorithm, such as ADAM. At each step, this entails sampling (1) a minibatch of data, (2) the variational posterior from $q(\boldsymbol{u}, f)$ so that one can calculate the expression inside the expectation. For classification, this expression reduces to a cross-entropy loss. This yields the correct loss in expectation, thereby defining a doubly stochastic variational inference (Titsias and Lázaro-Gredilla, 2014) algorithm for scalable training.

Variational inference with interdomain inducing variables. An alternative to using $\boldsymbol{u} = \boldsymbol{f}(\boldsymbol{z})$ as inducing variables is to consider *interdomain inducing variables* such as $u_i = \langle f, \psi_i \rangle$. Provided that the ψ_i are chosen wisely, this approach can results in order of magnitude speed ups (Hensman, Durrande, et al., 2017). For the problem at hand, it is natural to use the kernel eigenvectors as inducing functions ϕ_i (Burt et al., 2020). It is worth noting that this construction is exactly the discrete counterpart to the *variational inference with spherical harmonics* method recently developed by Dutordoir et al. (2020) in the continuous case.

4 Experiments

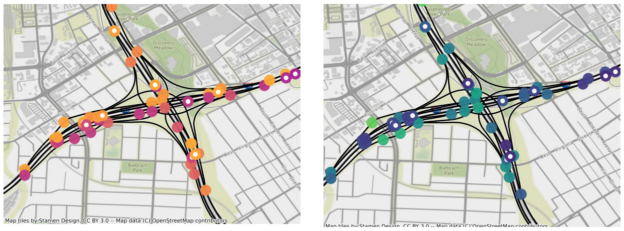
This section is dedicated to examples that illustrate the relevance of the proposed approach. Full experimental details are provided in Appendix A.


60
40
20
10
8
6
4
Sunnyvale
SJC
Burbank
Cambrian Park
Map tiles by Stamen Design, CC BY 3.0 -- Map data (C) OpenStreetMap contributors
Map tiles by Stamen Design, CC BY 3.0 -- Map data (C) OpenStreetMap contributors

(a) Mean

(b) Standard deviation

Figure 3: Traffic flow speed (mph) interpolation over a graph of San Jose highways. Colored circles represent graph nodes, black lines represent edges and colored circles with a white point in the middle represent training data. The standard deviation color bar is clipped: 10% most variable points are painted in yellow.



(a) Mean

(b) Standard deviation

Figure 4: Traffic flow speed (mph) interpolation over a particular road junction on a graph of San Jose highways. Colored circles represent graph nodes, black lines represent edges and colored circles with a white point in the middle represent training data.

4.1 Probabilistic graph interpolation of traffic data

To illustrate the flexibility of graph GPs in settings where other GP models may not be easy to apply, we consider probabilistic interpolation of traffic congestion data of highways in the city of San Jose, California. The graph of the road network is obtained from OpenStreetMap (OSM, 2017), and it is pre-processed to associate to each edge a weight equal to its inverse weight and by removing hanging nodes. We obtain traffic congestion data from the *California Performance Measurement System* database (Chen et al., 2001), which provides traffic flow speed in miles per hour at different locations, which we add as additional nodes onto the graph. Note that both travel directions on a highway are considered separate roads and thus form separate edges—this can be seen in Figure 4.

This process yields a graph with $1,016$ nodes and $1,173$ edges. Traffic flow speed is available at 325 nodes. We use 250 of these nodes, chosen at random, as training data, and the remainder as test data. The data is normalized by subtracting the mean over the training set and dividing the result by its standard deviation. For output values, we examine traffic congestion on Monday at 17:30 in the afternoon.

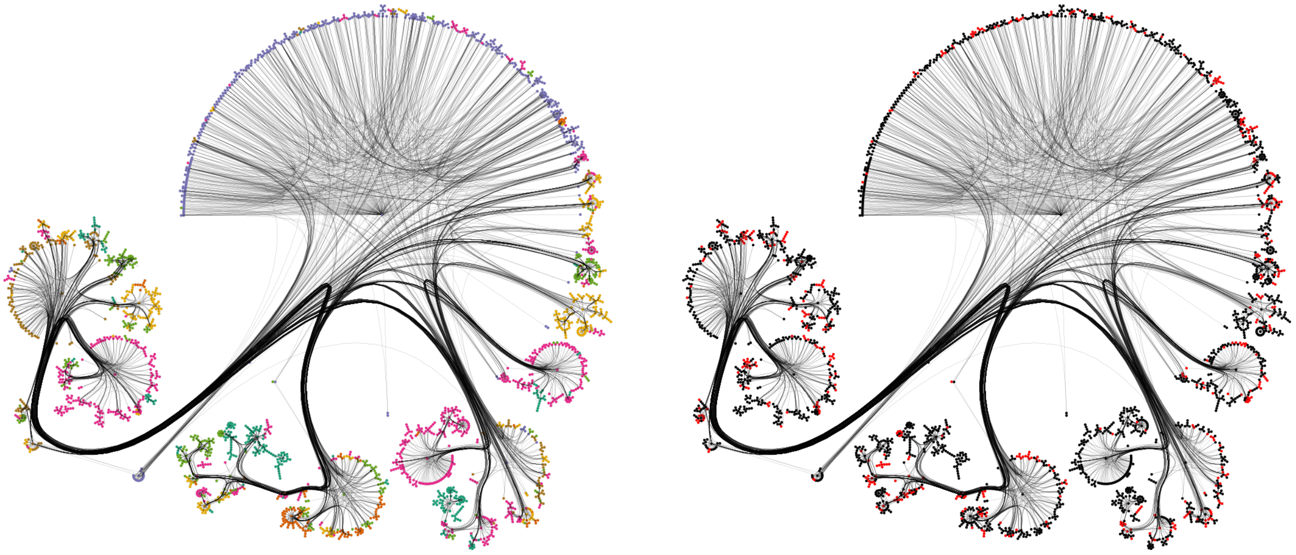
We train the Matérn graph GP model, whose kernel is approximated using 500 eigenpairs of the graph Laplacian. During training, we optimize the kernel hyperparameters κ (length scale), σ^2 (variance) and ν (smoothness), as well as the Gaussian likelihood error variance ς^2.

Figure 3 shows the predicted mean traffic on the entire graph, along with the standard deviation. This illustrates that global dependencies are captured by the model in different regions of the graph. In particular, the posterior GP's standard deviation increases for nodes for which traffic sensors are further away.

To illustrate local dependencies, Figure 4 details the model prediction at a particular road junction. It can be seen that despite being very close in the Euclidean distance, two locations on opposite sides of the road can have associated predictions that differ considerably. This shows that the graph GP incorporates the structure of the graph into its predictions, and allows them to differ when the graph distance is large.

4.2 Multi-class classification in a scientific citation network

To illustrate graph GPs in a more complex non-conjugate setting, we consider multi-class classification



(a) Ground truth

(b) Prediction errors. Red nodes are misclassified.

Figure 5: Topic classification for the *Cora* citation network, using the citation graph.

on the largest connected component of the *Cora* citation network. This yields a data set consisting of $2,485$ nodes corresponding to scientific publications, with $5,069$ edges corresponding to citations. We choose 140 nodes at random for training. Each node is labeled according to one of seven classes, which are different scientific topics. We employ only the graph and labels as part of the classification problem.

For the model, we use a multi-output Matérn graph GP where each output dimension is given using a separable kernel, yielding a GP $f : V \to \mathbb{R}^7$. We employ a categorical likelihood, with class probabilities given by applying a robust max function (Appendix A) to the GP. We approximate the graph kernel using 500 eigenpairs of the graph Laplacian, and train the model using variational inference in GPflow (Matthews et al., 2017; van der Wilk et al., 2020).

Performance is illustrated in Figure 5. The fraction of misclassified nodes is small, showing that in this data set, a paper's topic can be accurately predicted from its works cited alone even in a low-data regime.

4.3 Comparison with diffusion kernels

We now compare performance of GPs induced by the graph Matérn and graph diffusion kernels on both of the examples considered. As the latter is a limit of the former with the smoothness parameter ν tending to infinity, we aim to showcase how the additional flexibility stemming from the variable smoothness parameter affects performance. We evaluate performance using mean squared error, using 250 training nodes with 75 test nodes for the traffic example, and 140 training nodes with 1000 test nodes for the citation network example. We repeat this 10 times to obtain average mean squared error. Results are presented in Table 1. This shows that use of the more flexible Matérn class provides an improvement in overall accuracy.

	Matérn kern.	Diffusion kern.
Traffic MSE	**1.37** (0.16)	1.84 (0.11)
Citation accuracy	**0.77** (0.02)	0.47 (0.02)

Table 1: Mean and standard deviation (in parentheses) of model performance for different graph GP kernels. Note that the standard deviation of the traffic flow speed is approximately 17.1 mph.

5 Conclusion

In this work, we study graph Matérn Gaussian processes, which are defined as analogous to the Matérn stochastic partial differential equations used in the Euclidean and Riemannian cases. We discuss a number of their properties, and provide scalable training algorithms by means of Gaussian Markov random fields, graph Fourier features and sparse GPs. We demonstrate their effectiveness on a simple graph interpolation problem, and on a multi-class graph classification problem. Compared to other graph kernels, the graph Matérn class is flexible, interpretable, and shown to perform well, mirroring the behavior of Euclidean and Riemannian Matérn kernels. We hope these techniques inspire new use of Gaussian processes by machine learning practitioners.

Acknowledgments

This research was partially supported by "Native towns", a social investment program of PJSC Gazprom Neft and by the Ministry of Science and Higher Education of the Russian Federation, agreements № 075-15-2019-1619 and № 075-15-2019-1620.

References

V. Adam, S. Eleftheriadis, A. Artemev, N. Durrande, and J. Hensman. Doubly sparse variational Gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pages 2874–2884, 2020. Cited on page 6.

M. A. Alvarez, L. Rosasco, and N. D. Lawrence. Kernels for vector-valued functions: A review. *arXiv preprint arXiv:1106.6251*, 2011. Cited on page 5.

M. Belkin and P. Niyogi. Convergence of Laplacian eigenmaps. In *Advances in Neural Information Processing Systems*, pages 129–136, 2007. Cited on page 5.

V. Borovitskiy, A. Terenin, P. Mostowsky, and M. P. Deisenroth. Matern Gaussian processes on Riemannian manifolds. In *Advances in Neural Information Processing Systems*, 2020. Cited on pages 1, 2, 4, 5.

D. Burago, S. Ivanov, and Y. Kurylev. A graph discretization of the Laplace-Beltrami operator. *arXiv preprint arXiv:1301.2222*, 2013. Cited on page 5.

D. R. Burt, C. E. Rasmussen, and M. van der Wilk. Variational Orthogonal Features. *arXiv preprint arXiv:2006.13170*, 2020. Cited on page 6.

C. Chen, K. Petty, A. Skabardonis, P. Varaiya, and Z. Jia. Freeway performance measurement system: mining loop detector data. *Transportation Research Record*, 1748(1):96–102, 2001. Cited on pages 7, 11.

N. Durrande, V. Adam, L. Bordeaux, S. Eleftheriadis, and J. Hensman. Banded matrix operators for Gaussian Markov models in the automatic differentiation era. In *International Conference on Artificial Intelligence and Statistics*, pages 2780–2789, 2019. Cited on pages 2, 6.

V. Dutordoir, N. Durrande, and J. Hensman. Sparse Gaussian Processes with Spherical Harmonic Features. *arXiv preprint arXiv:2006.16649*, 2020. Cited on page 6.

L. C. Evans. *Partial Differential Equations.* American Mathematical Society, 2010. Cited on page 2.

A. Feragen, F. Lauze, and S. Hauberg. Geodesic exponential kernels: When curvature and linearity conflict. In *Conference on Computer Vision and Pattern Recognition*, 2015. Cited on page 1.

I. S. Gradshteyn and I. M. Ryzhik. *Table of Integrals, Series, and Products.* Academic Press, 7th edition, 2014. Cited on page 2.

A. Grigoryan. *Heat Kernel and Analysis on Manifolds*, volume 47. American Mathematical Society, 2009. Cited on page 2.

J. Hensman, N. Durrande, and A. Solin. Variational Fourier features for Gaussian processes. *Journal of Machine Learning Research*, 18:151–202, 2017. Cited on page 6.

J. Hensman, N. Fusi, and N. D. Lawrence. Gaussian processes for big data. In *Uncertainty in Artificial Intelligence*, 2013. Cited on page 6.

R. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete input spaces. In *Proceedings of the International Conference on Machine Learning*, pages 315–322, 2002. Cited on page 2.

N. M. Kriege, F. D. Johansson, and C. Morris. A survey on graph kernels. *Applied Network Science*, 5(1):1–42, 2020. Cited on page 3.

M. Lifshits. *Lectures on Gaussian Processes.* Springer, 2012. Cited on page 2.

F. Lindgren, H. Rue, and J. Lindström. An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(4):423–498, 2011. Cited on pages 1, 2, 5.

A. Mallasto and A. Feragen. Wrapped Gaussian process regression on Riemannian manifolds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5580–5588, 2018. Cited on page 1.

A. G. d. G. Matthews, M. van der Wilk, T. Nickson, K. Fujii, A. Boukouvalas, P. León-Villagrá, Z. Ghahramani, and J. Hensman. GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research*, 18(40):1–6, 2017. Cited on pages 8, 12.

Y. C. Ng, N. Colombo, and R. Silva. Bayesian semi-supervised learning with graph Gaussian processes. In *Advances in Neural Information Processing Systems*, pages 1683–1694, 2018. Cited on page 2.

OpenStreetMap contributors. Planet dump retrieved from https://planet.osm.org. `https://www.openstreetmap.org`, 2017. Cited on pages 7, 11.

A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, 2008. Cited on page 6.

C. E. Rasmussen and C. K. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006. Cited on pages 1, 2.

H. Rue and L. Held. *Gaussian Markov Random fields: Theory and Applications*. CRC Press, 2005. Cited on pages 1, 2, 4, 6.

D. Sanz-Alonso and R. Yang. The SPDE approach to Matérn fields: Graph representations. *arXiv preprint arXiv:2004.08000*, 2020. Cited on page 5.

A. J. Smola and R. Kondor. Kernels and regularization on graphs. In *Learning Theory and Kernel Machines*, pages 144–158. Springer, 2003. Cited on pages 3, 4.

M. L. Stein. *Interpolation of Spatial Data: Some Theory for Kriging*. Springer Science & Business Media, 1999. Cited on page 2.

M. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *Artificial Intelligence and Statistics*, 2009. Cited on page 6.

M. Titsias and M. Lázaro-Gredilla. Doubly stochastic variational Bayes for non-conjugate inference. In *International Conference on Machine learning*, pages 1971–1979, 2014. Cited on page 6.

M. J. Urry and P. Sollich. Random walk kernels and learning curves for Gaussian process regression on random graphs. *The Journal of Machine Learning Research*, 14(1):1801–1835, 2013. Cited on page 4.

M. van der Wilk, V. Dutordoir, S. John, A. Artemev, V. Adam, and J. Hensman. A Framework for interdomain and multioutput Gaussian processes. *arXiv preprint arXiv:2003.01115*, 2020. Cited on pages 8, 12.

A. Venkitaraman, S. Chatterjee, and P. Handel. Gaussian processes over graphs. In *International Conference on Acoustics, Speech and Signal Processing*, pages 5640–5644. IEEE, 2020. Cited on page 3.

S. V. N. Vishwanathan, N. N. Schraudolph, R. Kondor, and K. M. Borgwardt. Graph kernels. *Journal of Machine Learning Research*, 11:1201–1242, 2010. Cited on page 2.

Z. Wang, C. Gehring, P. Kohli, and S. Jegelka. Batched large-scale bayesian optimization in high-dimensional spaces. In *International Conference on Artificial Intelligence and Statistics*, pages 745–754. PMLR, 2018. Cited on page 6.

P. Whittle. Stochastic processes in several dimensions. *Bulletin of the International Statistical Institute*, 40(2):974–994, 1963. Cited on page 2.

J. T. Wilson, V. Borovitskiy, A. Terenin, P. Mostowski, and M. P. Deisenroth. Efficiently sampling functions from Gaussian process posteriors. In *International Conference on Machine Learning*, 2020. Cited on page 2.

Y.-C. Zhi, Y. C. Ng, and X. Dong. Gaussian processes on graphs via spectral kernel learning. *arXiv preprint arXiv:2006.07361*, 2020. Cited on page 3.

A Appendix: experimental details

A.1 Probabilistic graph interpolation of traffic data

Here we consider the problem of interpolating traffic congestion data over a road network, consisting of San Jose highways. We build the road network, a weighted graph, from the OpenStreetMap data (OpenStreetMap contributors, 2017).

We obtain traffic congestion data from the *California Performance Measurement Systems* database (Chen et al., 2001). This system maps sensor location and time & date to (among other factors) traffic flow speed in miles per hour measured by the given sensor at a particular time and date. We chose the specific date and time to be 17:30 on Monday, the 2^{th} of January, 2017. We bind the traffic congestion data to the graph by adding additional nodes that subdivide existing edges of the graph at the location of the measurement points. The edge weights are assigned to be equal to the inverse travel distances between the corresponding nodes. When we encounter two nodes connected by multiple edges, we remove all but one edge between them, and set its weight to be the inverse mean travel distance between the pair of nodes. After this, the graph is processed by (1) twice subsequently removing hanging nodes, and (2) removing two other nodes located particularly far away from data. We end up with 325 labeled nodes on a sparse connected graph with 1016 nodes and 1173 edges.

For the model, we employ a GP with a graph Matérn kernel and the graph diffusion kernel. Both kernels are approximated using 500 eigenpairs of the graph Laplacian. The eigenpairs required to build graph kernels are computed using TensorFlow's eigenvalue decomposition (TF.LINALG.EIGH) routine and are thus exact up to a numerical error arising from the floating point arithmetic. We train by optimzing the kernel hyperparameters κ (length scale), σ^2 (variance) and for the case of the graph Matérn kernel, ν (smoothness), as well as the Gaussian likelihood variance ς^2. We use the following values for initialization: $\nu = 3/2$ for the graph Matérn kernel, $\kappa = 3$, $\sigma^2 = 1$, $\varsigma^2 = 0.1$. We randomly choose 250 nodes with traffic flow rates as training data. The likelihood is optimized for 20000 iterations with the ADAM optimizer, with learning rate set to the TensorFlow default value of 0.001.

We repeat the experiment 10 times. The average MSE over the 10 experiments and its standard deviation are presented in Table 1 for the graph Matérn kernel and for the graph diffusion kernel. The smoothness parameter ν after optimization varies across the runs. Its mean is 1.8 and the standard deviation is 0.9. One of these ten runs is visualized in Figures 3 and 4.

A.2 Multi-class classification in a scientific citation network

Here we consider multi-class classification on the largest connected component of the *Cora* citation network.

There are 2708 nodes corresponding to scientific publications in the *Cora* citation network. Usually, each node of this graph is attributed by a 0/1-valued word vector indicating the absence/presence of the corresponding word from the dictionary of 1433 unique words in the abstract—we discard this portion of data to focus on other problem aspects.[3] Two nodes are connected by an edge if one paper cites another. The edges are unweighted and undirected, and there are 5429 edges overall. The nodes are labeled into one of seven classes. We concentrate on the largest connected component of the Cora graph, which has 2485 nodes and 5069 edges.

We consider a Gaussian process $V \to \mathbb{R}^7$ with independent components as a latent process for the classification task. The process is then combined with the robust max function $\boldsymbol{\varphi}(\cdot)$ to yield a vector of probabilities over the classes. Let $\boldsymbol{f} = (f_1, .., f_7)$ denote the values of 7-dimensional latent process. Then we have

$$\boldsymbol{\varphi}(\boldsymbol{f}) = (\varphi_1(\boldsymbol{f}), .., \varphi_7(\boldsymbol{f})) \qquad \varphi_c(\boldsymbol{f}) = \begin{cases} 1 - \varepsilon, & \text{if } c = \arg\max_c f_c \\ \varepsilon/6, & \text{otherwise,} \end{cases} \tag{23}$$

where ε is fixed to be 10^{-3}. For the likelihood, we employ the categorical distribution. We use the graph Matérn kernel and the graph diffusion kernel for the prior. Both kernels are approximated using 500 eigenpairs

[3] It's also possible to use a separable kernel given by the product of a graph Matérn kernel and a Euclidean kernel operating on word vectors, but this model is unlikely to perform well due to usual GP difficulties in high dimension. Incorporating this portion of the data effectively therefore necessitates additional modeling considerations, so we do not focus on it here.

of the graph Laplacian. The eigenpairs required are computed using TensorFlow eigenvalue decomposition (TF.LINALG.EIGH) and are thus exact up to a numerical error arising from use of floating point arithmetic. We use 140 random nodes as a training set and predict labels for the remaining 2345 nodes.

For the variational GP, we employ the standard SVGP model of GPflow (Matthews et al., 2017; van der Wilk et al., 2020). We fix the the coordinates of inducing points, which are located on the graph, to the data locations, and do not optimize them. We restrict the covariance of the inducing distribution to a diagonal form. The inducing mean is initialized to be zero and the inducing covariance is initialized to be the identity. The prior hyperparameters are initialized to be $\nu = 3$ for the Matérn kernel, $\kappa = 5$, $\sigma^2 = 1$. The variational objective is optimized for 20000 iterations with the ADAM optimizer whose learning rate set to 0.001. Convergence is usually achieved in a significantly lower number of iterations. The SVGP batch size is equal to the number of training data points, and whitening is enabled.

We repeat the experiment 10 times. Each time, we additionally randomly choose a test set of 1000 nodes. The average (over the 10 experiments) accuracy (fraction of the correctly classified points) and its standard deviation are presented in Table 1 for the graph Matérn kernel and for the graph diffusion kernel. The value of the smoothness parameter ν after optimization varies considerably across runs. The mean ν is 3.2 and the standard deviation is 2.7. The results of one of said ten runs are visualized on Figures 6, 7, 8. Note that Figures 6 and 8 repeat Figure 5 (a), (b) at a higher resolution.

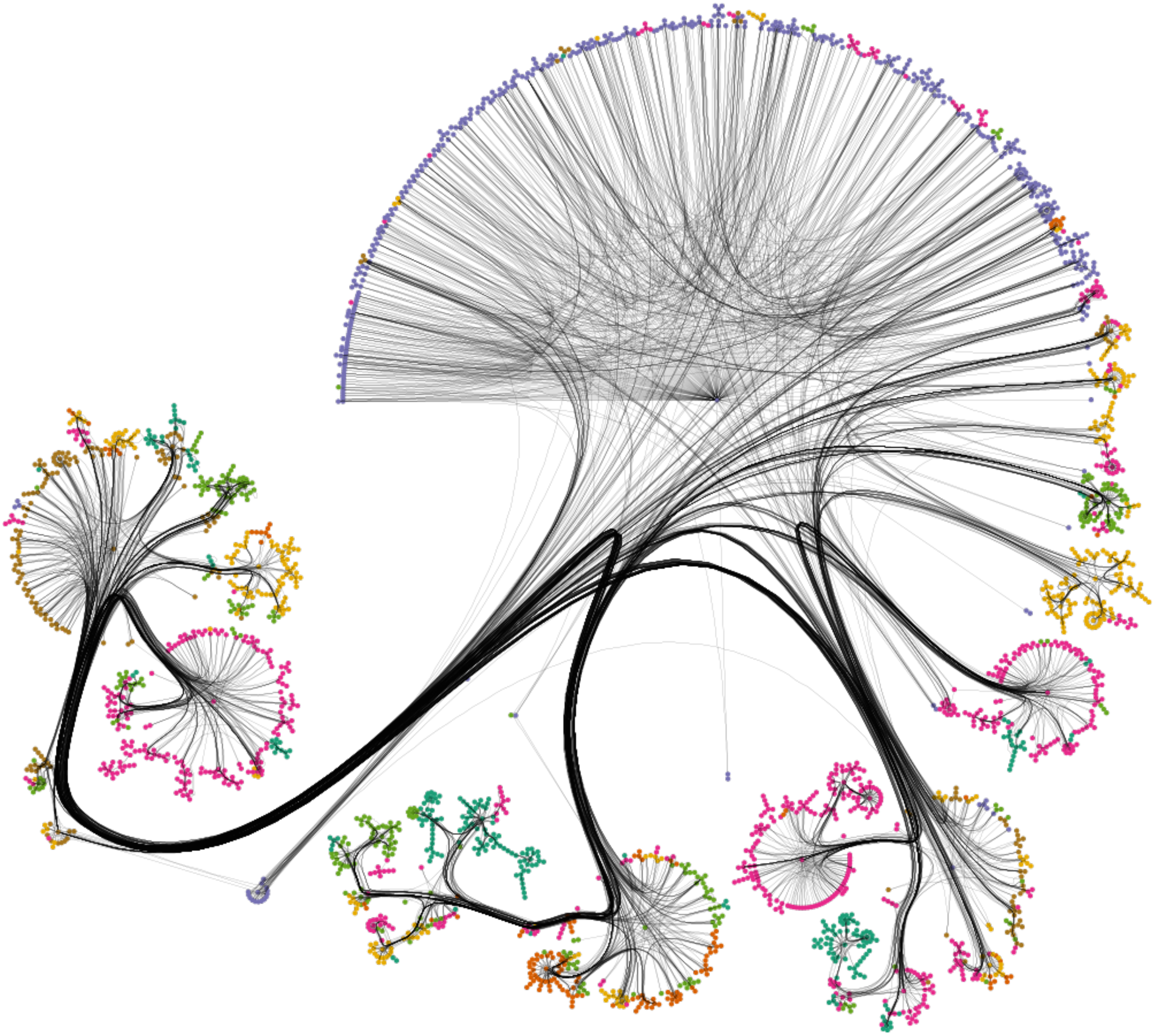


Figure 6: Topic classification for the *Cora* citation network, using the citation graph. Ground truth.

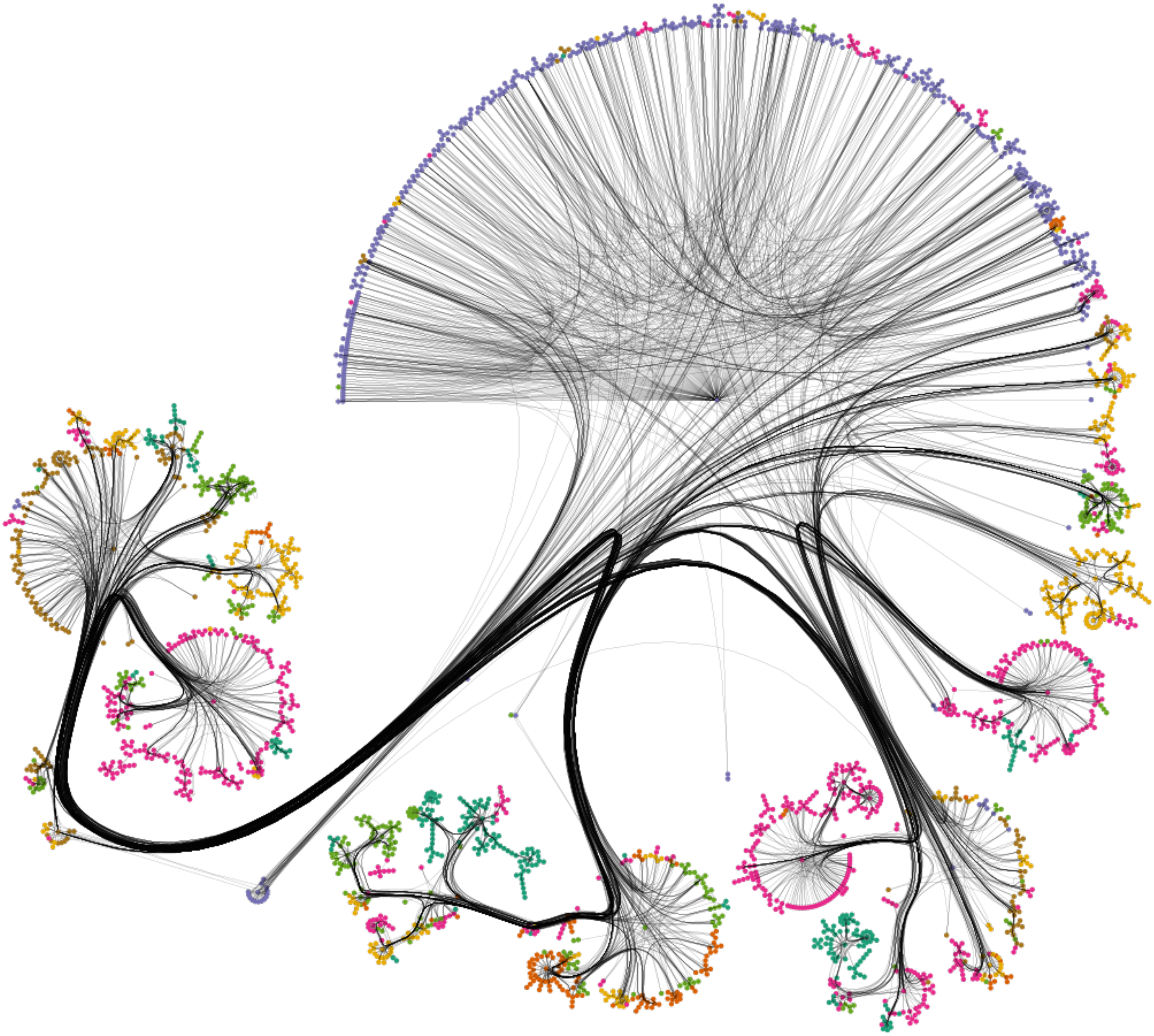


Figure 7: Topic classification for the *Cora* citation network, using the citation graph. Prediction.

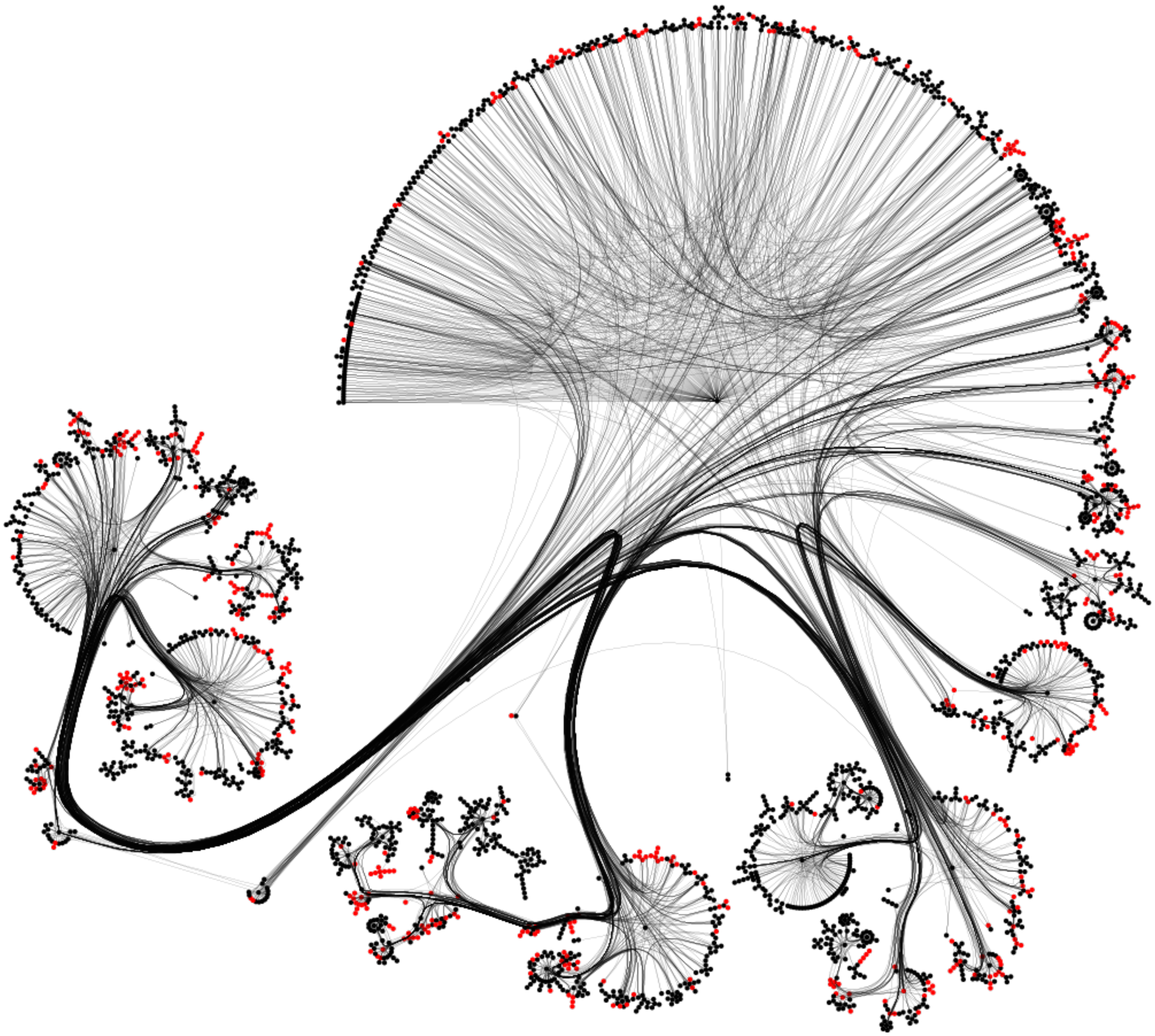


Figure 8: Topic classification for the *Cora* citation network, using the citation graph. Prediction errors. Red nodes are misclassified.