Non-separable Spatio-temporal Graph Kernels via SPDEs

Alexander Nikitin[1] **ST John**[1] **Arno Solin**[1] **Samuel Kaski**[1, 2]

[1]Finnish Center for Artificial Intelligence FCAI, Department of Computer Science, Aalto University, Finland
[2]Department of Computer Science, University of Manchester, UK

Abstract

Gaussian processes (GPs) provide a principled and direct approach for inference and learning on graphs. However, the lack of justified graph kernels for spatio-temporal modelling has held back their use in graph problems. We leverage an explicit link between stochastic partial differential equations (SPDEs) and GPs on graphs, introduce a framework for deriving graph kernels via SPDEs, and derive non-separable spatio-temporal graph kernels that capture interaction across space and time. We formulate the graph kernels for the stochastic heat equation and wave equation. We show that by providing novel tools for spatio-temporal GP modelling on graphs, we outperform pre-existing graph kernels in real-world applications that feature diffusion, oscillation, and other complicated interactions.

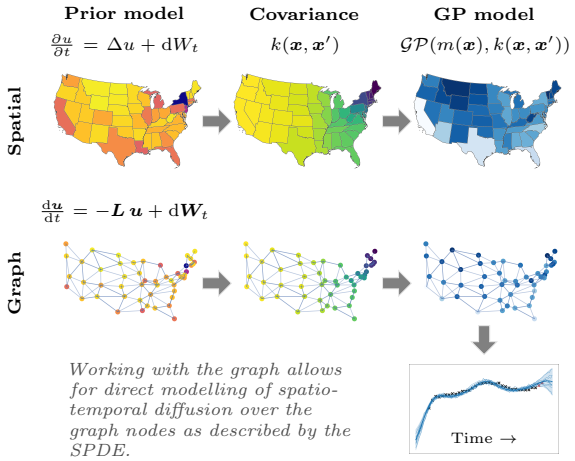


Figure 1: We start with an unknown process on a particular domain with an SPDE model, discretize it, and derive the GP covariance function from this SPDE. Finally, we use GP inference in the spatial or spatio-temporal model from discrete graph-structured observations.

1 INTRODUCTION

Contemporary machine learning typically leverages structure within the modelling task or application. Often, the structure can be represented as a graph, where the objects and relations are represented as nodes and edges. Such tasks can be found, for example, within chemoinformatics (Tsuda, 2011), social network analysis (Campbell et al., 2013), and anomaly detection (Akoglu et al., 2015). The abundance of graph-structured problems has led to the invention of new methods for graph-structured data, including graph neural networks (GNNs, Scarselli et al., 2008) and spectral methods (Chung and Graham, 1997) for capturing structure across space and time.

In this work, we focus on graph-based prediction of temporal signals on graphs. In other words, the goal is to approximate spatio-temporal processes on graphs. Many real-life problems belong to this class of tasks, including traffic prediction (Jiang and Luo, 2021), epidemiological modelling (Keeling and Eames, 2005), and analysis of dynamics in social networks (Carrington et al., 2005). Recently, a number of approaches have been developed for temporal machine learning problems on graphs, including the works by Li et al. (2017), Yu et al. (2018), and Zhao et al. (2019).

We approach these graph-based learning problems using Gaussian processes (GPs, Rasmussen and Williams, 2006). GPs are widely used tools in statistical machine learning, where including prior knowledge is desired. Their probabilistic treatment allows for convenient inference machinery and enables capturing the uncertainties associated with predictions. However, it is not

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

always apparent how to encode relevant prior knowledge into the GP prior on graphs. In the continuous domain, bringing in prior knowledge from stochastic partial differential equations (SPDEs) has enabled constructing useful kernels, including the Matérn family of kernels (Whittle, 1963; Lindgren et al., 2011), resulting in improvements in quality and performance in many tasks. In this work, we ask whether analogous constructs can be made for graphs and whether they improve the performance of GPs on spatio-temporal graph problems. We hypothesize that because many physical processes follow differential equations, deriving kernels from their stochastic counterparts will allow incorporating physical priors in GP models.

Apart from challenges in the model specification for graph-based GPs, the usual drawbacks also apply: closed-form inference is only possible for conjugate observation models, and a naïve treatment results in a cubic computational scaling with respect to the number of observations (Rasmussen and Williams, 2006). This means that vanilla GPs are not particularly suitable for large-scale problems. However, a number of modern methods exist to overcome this issue, including sparse Gaussian processes using inducing points (see Liu et al., 2020, for a recent review). The development of SPDE-based approaches to Gaussian processes has led to several practical solutions. One of those is a popular R package for approximate Bayesian inference: R-INLA (Lindgren et al., 2015) that uses SPDEs to sample from spatial models. The approach we propose in this paper allows extending this framework to graph problems, and it is illustrated in Fig. 1.

The contributions of this paper: *(i)* We extend methods from spatial statistics and, hence, link GPs to SPDEs for temporal signals on graphs. *(ii)* We apply this approach to spatio-temporal modelling on graphs and consider the analogs of well-known SPDEs from spatial statistics, which allows us to derive non-separable spatio-temporal kernels on graphs. *(iii)* We empirically evaluate these kernels and show their effectiveness on both synthetic data sets and applied machine learning problems: prediction of the distribution of chickenpox and COVID-19 epidemic.

2 RELATED WORK

The connection between Gaussian processes and stochastic (partial) differential equations was established by Whittle (1963). It resulted in a number of works in spatial statistics, most notably the work by Lindgren et al. (2011), the library R-INLA (Lindgren et al., 2015), and by Solin (2016) in the spatio-temporal context. These methods have been applied in many domains, for example, animal movement prediction (Hooten et al., 2017), geographical information system modelling (Burrough et al., 2015), and geostatistical analysis (Moraga et al., 2017).

To apply SPDE approaches to graphs, we need to introduce spectral graph theory, which has a long history of research and applications (e.g., Chung and Graham, 1997; Von Luxburg, 2007). These works typically consider the spectrum of the graph Laplacian and derive properties that can be helpful for different tasks on graphs. Belkin and Niyogi (2003) used spectral graph theory to develop invariant embedding maps. These embedding maps then showed good results in dimensionality reduction (Yan et al., 2007). Spectral graph theory is linked to GPs on graphs by considering suitable spatial (pseudo-)differential operators and their associated kernels (covariance functions).

A number of graph kernels were developed using the methods from spectral graph theory. Smola and Kondor (2003) proposed the construction of kernels on graphs by introducing random walk and heat kernels on graphs, evaluated their performance, and derived necessary properties of the kernels. Recent work on graph approaches to Matérn fields considers graph discretization and theoretically compares the covariances produced by discrete approximation and Matérn kernels (Sanz-Alonso and Yang, 2020). Graph representation learning has gained popularity because of its large number of applications (Chami et al., 2020). Kernel methods, such as kernel ridge regression, for prediction on graphs were proposed by Romero et al. (2016). This approach was extended to spatio-temporal signals on dynamic graphs (Romero et al., 2017). Matérn graph kernels were presented by recent works: Matérn kernels on manifolds (Borovitskiy et al., 2020), and, then, on graphs (Borovitskiy et al., 2021).

We extend and generalize this approach with the SPDE framework for graph kernels and then go beyond spatial graph kernels to the spatio-temporal domain.

3 BACKGROUND

Gaussian processes are a non-parametric machine learning paradigm, in which we model the target function as a stochastic process, whose evaluation at any finite set of points has a joint Gaussian distribution (Rasmussen and Williams, 2006). A Gaussian process $f(\boldsymbol{x}) \sim \mathcal{GP}(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}'))$ is defined by its mean function $m(\boldsymbol{x})$ and covariance function $k(\boldsymbol{x}, \boldsymbol{x}')$ (Bishop, 2006). The covariance function is often called a 'kernel'. The kernel encapsulates prior knowledge, and defining a good kernel is one of the key ingredients and challenges of setting up the GP model (Duvenaud, 2014). GPs can be extended to vector-valued functions using multioutput GPs (Álvarez et al., 2012). Kernels then

become matrix-valued: $\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}') : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{m \times m}$, for n-dimensional inputs and m-dimensional outputs. Standard GP toolchains include various kernels on continuous domains (e.g., periodic, polynomial, and Matérn family). However, application of GPs to other domains is often restricted by the lack of principled kernels.

3.1 Stochastic Partial Differential Equations

In spatial statistics, Gaussian random fields can be represented as solutions to an SPDE. The physical sense of an underlying process can be included in GPs in a natural way by forming the prior covariance function as a solution to an SPDE (Särkkä, 2011). Many widely used GP kernels can be derived from the corresponding SPDEs. For example, the Matérn kernel family for multidimensional GPs can be derived from underlying SPDEs (Whittle, 1963).

Stochastic partial differential equations (SPDEs) generalize partial differential equations via introducing random forces (noise) to some terms and coefficients, similarly to how stochastic differential equations generalize ordinary differential equations (Oksendal, 2013; Gardiner, 2004; Särkkä and Solin, 2019). SPDEs are applied in various fields, including physics, signal processing, and machine learning (Schottl, 1997).

3.2 Spectral Graph Theory

Spectral graph theory allows us to derive graph properties by operating with the spectrum of its Laplacian matrix (a.k.a. graph Laplacian), which will allow us to define partial differential equations (PDEs) on graphs.

Definition 3.1. *The graph Laplacian $\boldsymbol{L}$ of a graph $G = (V, E)$ is the matrix*

$$\boldsymbol{L} = \boldsymbol{D}_W - \boldsymbol{W}, \tag{1}$$

where $\boldsymbol{W}$ is the matrix of the edge weights and $\boldsymbol{D}_W = \mathrm{diag}(w_i)$ is the diagonal matrix of the accumulated weights $w_i = \sum_{j:(i,j)\in E} w_{ij}$. In the unweighted case, $\boldsymbol{D}_W$ is the degree matrix and $\boldsymbol{W}$ is the adjacency matrix.

The graph Laplacian can be normalized to have unit diagonal entries. All derivations in this article apply to both normalized and non-normalized Laplacians in weighted and unweighted graphs. In practice, the choice between normalized and unnormalized Laplacians depends on the graph structure and should be based on the particular modeling task.

4 SPATIO-TEMPORAL KERNELS ON GRAPHS VIA SPDEs

We consider a spatio-temporal prediction problem on a graph $G = (V, E)$ given a data set $\{(x_i, y_i)\}_{i=1}^N$, with features $x_i \in D$, where the domain $D = V \times \mathbb{R}$ describes graph vertex and time. We focus on the regression task, where y_i are the targets from the real numbers $\mathbb{R}$. We aim to approximate a function f such that $f : D \to \mathbb{R}$. The methods we propose are applicable for classification problems $f : D \to C$ for a finite set of labels C as well.

Notation. If not stated otherwise, we denote $f(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}$ a continuous spatial function, $\boldsymbol{v} \in \mathbb{R}^{|V|}$ a spatial function on a graph (i.e., a vector whose components are the function values at the vertices), and $\boldsymbol{u}(t) : \mathbb{R} \to \mathbb{R}^{|V|}$ a spatio-temporal function on a graph. We will denote $w(\boldsymbol{x})$ spatial white noise[1], $\boldsymbol{w}$ a $|V|$-dimensional white noise[2], and $\mathrm{d}\boldsymbol{W}_t$ the differential of $|V|$-dimensional Brownian motion. $\dot{\boldsymbol{W}}_t = \frac{\mathrm{d}\boldsymbol{W}_t}{\mathrm{d}t}$ is the *formal* temporal derivative of Brownian motion[3], and $\boldsymbol{W}_t$ is Brownian motion.

4.1 SPDE Framework for Graph Kernels

Our framework for deriving graph kernels via SPDEs consists of the following high-level steps: *(i)* define an SPDE using prior knowledge about the underlying process, *(ii)* convert it to a graph counterpart, *(iii)* solve graph counterpart, and *(iv)* derive corresponding mean and covariance function of GP on graph. Next, we will provide specific examples and use this framework to derive novel kernels.

Many PDEs are defined using the Laplace operator Δ. In Euclidean space, it is defined as $\Delta = \sum_{i=1}^n \frac{\partial^2}{\partial x_i^2}$. On graphs, we define as the corresponding operator the negative graph Laplacian $-\boldsymbol{L}$. This is justified by considering a lattice approximation of the real plane and ensuring that this operator represents exactly the discretization of Δ (Wardetzky, 2020).

We illustrate the approach by considering the analogue of Laplace's equation on graphs. Laplace's equation, $\Delta f(\boldsymbol{x}) = 0$, is a second-order partial differential equation relevant across physics, for example, in fluid dynamics and electromagnetism. Its stochastic counterpart, known as Laplace's stochastic partial differential equation, is $\Delta f(\boldsymbol{x}) = w(\boldsymbol{x})$. For the analogue of Laplace's SPDE on a graph, we replace Δ with $-\boldsymbol{L}$

[1] I.e., $\mathbb{E}[w(\boldsymbol{x})] = 0$, $\mathrm{Cov}[w(\boldsymbol{x}), w(\boldsymbol{x}')] = \delta(\boldsymbol{x} - \boldsymbol{x}')$.

[2] I.e., $\mathbb{E}[\boldsymbol{w}] = \boldsymbol{0}$ and $\mathbb{E}[\boldsymbol{w}\boldsymbol{w}^\intercal] = \boldsymbol{I}$.

[3] Even though Brownian motion is nowhere differentiable, assuming its derivative to be temporal white noise is the standard way of treating it in SDE literature as discussed by Särkkä and Solin (2019).

and arrive at

$$-\boldsymbol{L}\boldsymbol{v} = \boldsymbol{w}. \tag{2}$$

Because the space we are considering is discrete and the graph Laplacian replaces the Laplace operator, the graph counterparts of SPDEs become multi-dimensional SDEs.

For a formal derivation using discrete calculus, one can consider $\boldsymbol{v}$ as a 0-cochain, as in Grady and Polimeni (2010). Within our work, it is enough to operate with $\boldsymbol{v}$ as a vector. Eq. (2) defines the notion of harmonic functions on graphs: in functions that are solutions to Eq. (2), the value at each node is given by the mean of adjacent nodes.

The solution $\boldsymbol{v}$ of Eq. (2) can be described as a Gaussian process (which in the discrete case, by definition, is a multivariate normal distribution):

$$\boldsymbol{v} \sim \mathcal{N}(\boldsymbol{0}, (\boldsymbol{L}^\top \boldsymbol{L})^+), \tag{3}$$

where $\boldsymbol{A}^+$ denotes the Moore–Penrose pseudoinverse of matrix $\boldsymbol{A}$ (Moore, 1920). This defines the Laplacian kernel on graphs, $\boldsymbol{K} = (\boldsymbol{L}^\top \boldsymbol{L})^+$.

Another example of the SPDE framework applied to graph kernels is the Matérn family of kernels. The Matérn kernel family is an essential kernel family for many applications of GPs. The isotropic (rotation and translation invariant) Matérn kernels are defined as (Rasmussen and Williams, 2006)

$$k_{\text{Matérn}}(r) = 2^{1-\nu}\gamma^\nu K_\nu(\gamma)/\Gamma(\nu), \quad \gamma = \sqrt{2\nu}r/\kappa, \tag{4}$$

where $r = \|\boldsymbol{x} - \boldsymbol{x}'\|$ is the Euclidean distance, κ is the lengthscale, and ν controls smoothness (samples from a GP with this kernel are $(\lceil\nu\rceil - 1)$-times differentiable). The Matérn kernel in $\mathbb{R}^d$ can be derived from this SPDE with the fractional Laplace operator (Lindgren et al., 2011):

$$\left(\frac{2\nu}{\kappa^2} - \Delta\right)^{\frac{\nu}{2}+\frac{d}{4}} f(\boldsymbol{x}) = w(\boldsymbol{x}). \tag{5}$$

This motivates us to use the fractional graph Laplacian:

Definition 4.1. *Fractional graph Laplacian* $\widetilde{\boldsymbol{L}}$*:*

$$\widetilde{\boldsymbol{L}} = \left(\frac{2\nu}{\kappa^2}\boldsymbol{I} + \boldsymbol{L}\right)^{\frac{\nu}{2}}, \tag{6}$$

where ν *and* κ *are positive scalar parameters.*

The fractional graph Laplacian is more flexible than $\boldsymbol{L}$ (for $\nu = 2$ and $\kappa \to \infty$, it coincides with $\boldsymbol{L}$). Additionally, for some combinations of hyperparameters, the fractional graph Laplacian possesses non-local properties (see Benzi et al. (2020)).

We can formulate Eq. (5) on graphs as

$$\widetilde{\boldsymbol{L}}\boldsymbol{v} = \boldsymbol{w}. \tag{7}$$

For a self-adjoint Laplacian $\boldsymbol{L}$ the covariance matrix is

$$\boldsymbol{K} = \left(\frac{2\nu}{\kappa^2}\boldsymbol{I} + \boldsymbol{L}\right)^{-\nu}, \tag{8}$$

and the solution of Eq. (7) is $\boldsymbol{v} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{K})$. These results were recently considered by Borovitskiy et al. (2021), and here we showed how they fit into a broader SPDE framework.

4.2 Separable Spatio-Temporal Kernels on Graphs

To obtain a kernel on the combination of two domains (here, graph and time), one can take kernels for each domain and combine them, commonly as a product: $k(\boldsymbol{x}, t; \boldsymbol{x}', t') = k_{\boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{x}')\, k_t(t, t')$, where $\boldsymbol{x}$ and $\boldsymbol{x}'$ belong to the spatial domain and t and t' to the temporal domain. The separability of a kernel is a choice made for convenience. An underlying process is not necessarily separable; an example of spatial covariance structure changing over time can be easily constructed: for instance, in spatio-temporal epidemiological modelling, the structure of the spatial covariance changes over time (e.g., travel restriction or development of a vaccine), thus it is not enough to separate spatial and temporal structure, and they have to be considered jointly.

4.3 Non-Separable Kernels

We describe non-separable spatio-temporal graph kernels using the SPDE framework. Let us consider two widely used parabolic and hyperbolic equations:

- Heat equation: $\frac{\partial f}{\partial t} = c\Delta f$,
- Wave equation: $\frac{\partial^2 f}{\partial t^2} = c^2\Delta f$.

We will define the stochastic counterparts of these equations on graphs, derive the corresponding covariance functions, and study them. The kernel derivations from these two SPDEs are formulated as propositions. The full derivations are in Appendices B and C.

4.4 Stochastic Heat Equation

The heat equation is written as $\frac{\partial f}{\partial t} = c\Delta f$. This PDE is called the *heat* equation because it is used in physics for modeling heat transfer on surfaces. If f is temperature, the Laplace operator on the equation's right-hand side determines the difference between the temperature at a certain point and its neighboring points on a surface.

We consider the discretization of this equation using the graph Laplacian. Let $\boldsymbol{u}(t)$ be a temporal signal on a graph which obeys the heat equation

$$\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} = -c\boldsymbol{L}\boldsymbol{u}, \tag{9}$$

which is now an ODE with the solution

$$\boldsymbol{u}(t) = e^{-c\boldsymbol{L}t}\boldsymbol{u}(0). \tag{10}$$

This solution is known as the heat kernel. This result plays an important role in graph theory. The positive semi-definite matrix $e^{-c\boldsymbol{L}t}$ in this equation can be used as a kernel on the graph. This kernel is related to graph random walks, which are used widely in graph algorithms (Lovász, 1993). To see this, consider a Taylor expansion of this kernel (assume $c = 1$ for simplicity): $e^{-\boldsymbol{L}t} = \sum_{k=0}^{+\infty} \frac{t^k e^{-t}}{k!}\boldsymbol{P}^k$, where $\boldsymbol{P} = \boldsymbol{D}^{-1}\boldsymbol{A}$ is the random walk matrix of the graph. In fact, this kernel defines continuous-time random walks, where each jump occurs in Poisson(1) time, similarly to the continuous case.

By adding temporal white noise, we can construct a heat SPDE on graphs:

$$\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} = -c\boldsymbol{L}\boldsymbol{u} + \sigma\dot{\boldsymbol{W}}_t. \tag{11}$$

Proposition 1. *The **stochastic heat equation kernel (SHEK)** on graphs can be defined by adding spatio-temporal white noise, or for convenient integration, as a* formal *derivative of the Wiener process $\dot{\boldsymbol{W}}_t$:*

$$\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} = -c\widetilde{\boldsymbol{L}}\boldsymbol{u} + \sigma\dot{\boldsymbol{W}}_t. \tag{12}$$

The solution is given by a Gaussian process:

$$\begin{aligned}
&\boldsymbol{u}(t) \sim \mathcal{GP}(\boldsymbol{\mu}(t), \mathrm{Cov}[\boldsymbol{u}(s), \boldsymbol{u}(t)]), \quad \textit{with} \\
&\boldsymbol{\mu}(t) = e^{-c\widetilde{\boldsymbol{L}}t}\boldsymbol{u}(0), \\
&\mathrm{Cov}[\boldsymbol{u}(t), \boldsymbol{u}(s)] = \frac{\sigma^2}{c} e^{-c\widetilde{\boldsymbol{L}}t - c\widetilde{\boldsymbol{L}}^\top s} \\
&\quad (e^{c(\widetilde{\boldsymbol{L}}+\widetilde{\boldsymbol{L}}^\top)\min(t,s)} - \boldsymbol{I})(\widetilde{\boldsymbol{L}} + \widetilde{\boldsymbol{L}}^\top)^{-1}.
\end{aligned} \tag{13}$$

Or, when the matrix $\widetilde{\boldsymbol{L}}$ is self-adjoint (the graph is undirected), as

$$\begin{aligned}
&\boldsymbol{\mu}(t) = e^{-c\widetilde{\boldsymbol{L}}t}\boldsymbol{u}(0), \\
&\mathrm{Cov}[\boldsymbol{u}(t), \boldsymbol{u}(s)] = \frac{\sigma^2}{2c}\left(e^{-c\widetilde{\boldsymbol{L}}|t-s|} - e^{-c\widetilde{\boldsymbol{L}}(t+s)}\right)\widetilde{\boldsymbol{L}}^{-1}.
\end{aligned} \tag{14}$$

The kernel is parameterized by diffusivity c, variance σ, and parameters of the fractional Laplacian ν and κ. We extend this result to the case of matrix-scaled white noise:

Proposition 2. ***SHEK with matrix-scaled noise.*** *A scaled extension of SHEK can be derived from a stochastic heat equation on graphs with a matrix-scaled white noise:*

$$\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} = -c\widetilde{\boldsymbol{L}}\boldsymbol{u} + \boldsymbol{\Sigma}\dot{\boldsymbol{W}}_t. \tag{15}$$

The solution for undirected graphs is defined by a Gaussian process with covariance matrix in the following form:

$$\mathrm{Cov}\left[\boldsymbol{u}(t), \boldsymbol{u}(s)\right] = \boldsymbol{P}^*\boldsymbol{C}(t,s)\boldsymbol{P}, \tag{16}$$

where $\boldsymbol{P}$ is a unitary matrix:

$$\boldsymbol{P}\widetilde{\boldsymbol{L}}\boldsymbol{P}^* = \mathrm{diag}(\lambda_1, \ldots, \lambda_{|V|}). \tag{17}$$

The matrix $\boldsymbol{P}$ exists because $\widetilde{\boldsymbol{L}}$ is normal and positive definite. $\boldsymbol{C}(t,s)$ is defined for $t \geq s$ as:

$$\begin{aligned}
&\boldsymbol{C}(t,s)_{i,j} = \frac{1}{c}\frac{(\boldsymbol{P}\boldsymbol{\Sigma}\boldsymbol{\Sigma}^\top\boldsymbol{P}^*)_{i,j}}{\lambda_i + \lambda_j} \\
&\times \left(\exp(-c\lambda_i|t-s|) - \exp(-c(\lambda_i t + \lambda_j s))\right).
\end{aligned} \tag{18}$$

For a directed graph, the stationary covariance matrix $\boldsymbol{C}_*$ can be found from the Lyapunov equation

$$\widetilde{\boldsymbol{L}}\boldsymbol{C}_* + \boldsymbol{C}_*\widetilde{\boldsymbol{L}}^\top = \boldsymbol{\Sigma}\boldsymbol{\Sigma}^\top. \tag{19}$$

4.5 Wave Equation

Spreading waveforms in physics, such as mechanical waves (e.g., sound waves or seismic waves) as well as light waves, are commonly described by the wave equation:

$$\frac{\mathrm{d}^2 f}{\mathrm{d}t^2} = c^2\Delta f. \tag{20}$$

This PDE can be considered on graphs as follows:

$$\frac{\mathrm{d}^2\boldsymbol{u}}{\mathrm{d}t^2} = -c^2\boldsymbol{L}\boldsymbol{u}. \tag{21}$$

The solution of this equation for undirected graphs has the form

$$\boldsymbol{u}(t) = \frac{1}{c}\sqrt{\boldsymbol{L}^+}\sin(c\sqrt{\boldsymbol{L}}t)\boldsymbol{P}\dot{\boldsymbol{u}}(0) + \cos(c\sqrt{\boldsymbol{L}}t)\boldsymbol{P}\boldsymbol{u}(0), \tag{22}$$

where $\boldsymbol{P}$ is defined by diagonalization of the Laplacian matrix: $\boldsymbol{L} = \boldsymbol{P}^{-1}\,\mathrm{diag}(\lambda_1, \ldots, \lambda_{|V|})\boldsymbol{P}$. The solution of the stochastic wave equation cannot be used directly as a GP kernel (as it was with the heat equation), but this result is necessary for the derivation of a GP kernel from stochastic wave equations on graphs.

4.6 Stochastic Wave Equation

In this section, we consider a stochastic extension of the wave equation on graphs. Adding stochasticity allows us to derive a GP kernel that will include prior information about wave nature of the underlying process.

Proposition 3. *The **stochastic wave equation kernel (SWEK)** on undirected graphs is defined by the second-order matrix differential equation*

$$\frac{\mathrm{d}^2\boldsymbol{u}}{\mathrm{d}t^2} = -c^2\widetilde{\boldsymbol{L}}\boldsymbol{u} + \sigma\dot{\boldsymbol{W}}_t, \tag{23}$$

and a solution to this equation for undirected graphs can be expressed by the Gaussian process:

$$\boldsymbol{u}(t) \sim \mathcal{GP}(\boldsymbol{\mu}, \mathrm{Cov}[\boldsymbol{u}(s), \boldsymbol{u}(t)], \quad \textit{with} \tag{24}$$

$$\boldsymbol{\mu}(t) = \frac{1}{c}\widetilde{\boldsymbol{L}}^{-\frac{1}{2}}\sin(c\sqrt{\widetilde{\boldsymbol{L}}}t)\boldsymbol{P}\dot{\boldsymbol{u}}(0) + \cos(c\sqrt{\widetilde{\boldsymbol{L}}}t)\boldsymbol{P}\boldsymbol{u}(0),$$

$$\mathrm{Cov}[\boldsymbol{u}(s), \boldsymbol{u}(t)] = \sigma^2\boldsymbol{\Theta}^{-2}\Big(\cos(\boldsymbol{\Theta}(t-s))\,\min(t,s) - \frac{1}{2}\cos(\boldsymbol{\Theta}\max(t,s))\sin(\boldsymbol{\Theta}\min(t,s))\boldsymbol{\Theta}^{-1}\Big),$$

where $\boldsymbol{\Theta} = c\sqrt{\widetilde{\boldsymbol{L}}}$ and $\boldsymbol{P}$ is defined by the diagonalization of the fractional Laplacian matrix:

$$\widetilde{\boldsymbol{L}} = \boldsymbol{P}^{-1}\widetilde{\boldsymbol{L}}_d\boldsymbol{P}. \tag{25}$$

We can observe that eigenvalues of the kernel oscillate with different frequencies. It makes this kernel potentially useful for periodic processes on graphs or discretizations of continuous periodic processes with a graph lattice.

The stochastic wave equation kernel can be extended to directed graphs similarly by solving the graph wave equation through the eigenvectors of $\widetilde{\boldsymbol{L}}$.

In a single-vertex graph, this kernel coincides with the product of Brownian motion kernel and cosine kernel.

4.7 Sampling in Graph GP Models

Many methods for sampling from GPs are available. A common method involves a Cholesky decomposition of the covariance matrix. The computational cost of sampling is a sum of $O(n^3)$ on Cholesky factorization and $O(n^2)$ (matrix-vector multiplication) for each sample, where n is the number of points.

Many other sampling approaches are based on the expansion of the random field (process) as a series of basis functions. Those methods include Karhunen–Loeve expansion using the eigenfunctions of a covariance function, circulant embedding methods based on Fourier basis (Graham et al., 2018), and a hierarchical matrix approximation of the covariance matrix (Feischl et al., 2018). The SPDE based approach (Lindgren et al., 2011) requires solving an SPDE using the finite element method, which approximates the GP with a Gaussian Markov random field (GMRF).

The SPDE framework allows for using the numerical solvers of SPDEs, for example, the Ruge–Stüben solver (Ruge and Stüben, 1987) for the spatial case or Euler–Maruyama (Higham et al., 2002) for the spatio-temporal case. This idea especially shines in the graph domain because the Laplacian matrix allows considering these equations as matrix equations; thus, it allows using all of the existing tools for numerical solutions of matrix equations. Moreover, in principle, continuous domains can be discretized via graph representations, and then numerical solvers can be used for sampling from GPs over continuous domains. We consider this opportunity as one more justification for the SPDE framework and leave its development for future works.

4.8 Visualizations of the Kernels

For a deeper understanding of the properties of the kernels and a more pictorial tour of their properties, we study visualizations of the kernels on a simple graph with different values for the kernel hyperparameters.

In Fig. 2, we show the visualizations of the proposed kernels on a toy model: a line graph of three vertices. The first row shows the temporal part of the covariance matrix (summed over the graph vertices at each time-point). The following three rows show mean (black) and samples (colored lines) as a function of time at each of the nodes, conditioned on $y(t=0) = (0, 0, 10)$, for different values of the hyperparameter c ($\nu = 3/2$, $\kappa = 1$). For the temporal component, SHEK shows the expected behavior for a diffusion process: covariance is high between nearby time points and decreases with the temporal gap, whereas SWEK shows periodic dependence between temporal components (periods of high covariance are alternating with periods of low covariance). Additionally, stochasticity accumulates with time resulting in higher covariance for later timestamps (analogously to Brownian motion).

5 EXPERIMENTS

To highlight practical applicability of the proposed methodology and benchmark against alternative approaches, we evaluate the described kernels on graph spatio-temporal applications: heat distribution over a one-dimensional line (Sec. 5.1), spreading of COVID-19 across the United States (Sec. 5.2), and spreading of chickenpox over Hungarian counties (Sec. 5.3).

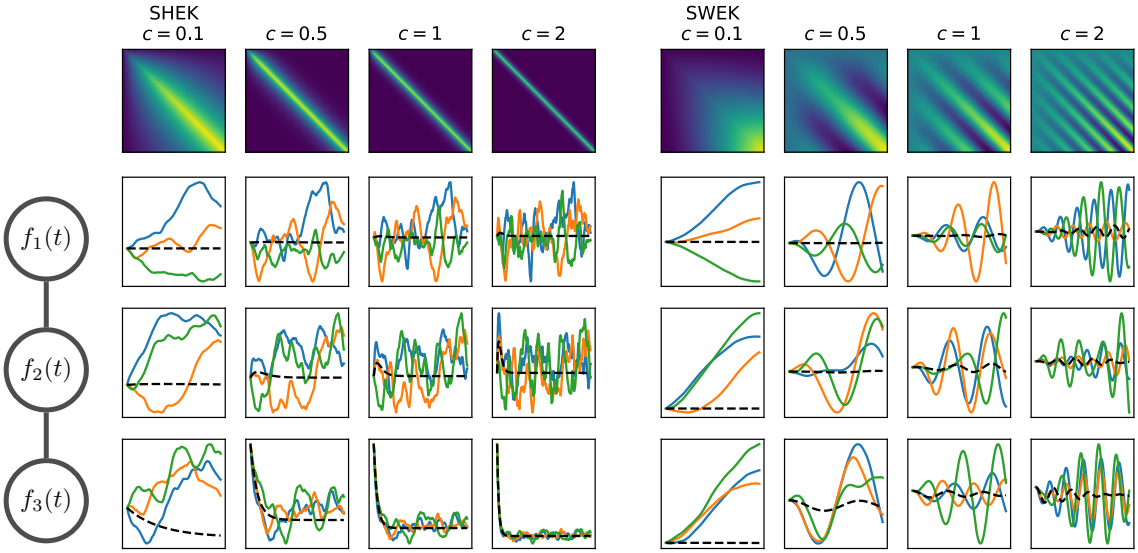


Figure 2: Temporal visualizations of SHEK (heat, left) and SWEK (wave, right) on a linear three-node graph. The first row shows the temporal part of the covariance matrix (summed over the graph vertices at each timepoint). The following three rows show mean (black) and samples (colored lines) as a function of time at each of the nodes, conditioned on $y(t = 0) = (0, 0, 10)$, for different values of the hyperparameter c.

We used Gaussian process regression with Gaussian likelihood as a method for fitting the models. We maximized the log marginal likelihood with respect to kernel hyperparameters (σ and c for SWEK and SHEK) with the L-BFGS optimizer (Liu and Nocedal, 1989). All the experiments were implemented using the GPflow (Matthews et al., 2017) library in Python.

Extrapolation. Our experiments were repeated for different training and testing intervals (we call them validation rounds) for a given data set using sliding window backtesting. If we denote number of training timepoints as N_{train}, number of testing timepoints as N_{test}, and by t^r the starting time for the r-th validation round, then sliding backtesting uses time interval $[t^r, \ldots, t^r + N_{\text{train}}]$ as the training data, and $[t^r + N_{\text{train}} + 1, \ldots, t^r + N_{\text{train}} + N_{\text{test}}]$ as test data.

Interpolation. To evaluate interpolation properties of the methods, we randomly selected ten percent of the time interval $[t^r, \ldots, t^r + N_{\text{train}} + N_{\text{test}}]$ as testing data set and used the rest for training.

Evaluation. In forecasting problems, modelling may result in a high variance of the generalization error across the validation rounds. Thus, the comparison of generalization error over the rounds by confidence intervals can be inaccurate. The generalization errors should be evaluated with a thorough point-wise comparison over each evaluation. To make statistically significant conclusions, we used the Diebold–Mariano test (Diebold and Mariano, 2002). The variance of the generalization error is also an important model property that shows the method's robustness over different validation rounds. We tested both interpolation and extrapolation performance and report mean absolute error (MAE_{int} and MAE_{ext}) or mean absolute percentage error (MAPE_{int} and MAPE_{ext}). We also report 95% confidence intervals over validation rounds.

5.1 Heat Transfer Data Set

As a simple spatio-temporal process, we considered heat transfer in 1D on a line. We parameterized the process with conductivity k, and considered a fundamental solution of the heat distribution process: $\Phi(x, t) = 5/(4\pi kt) \exp(-x^2/4kt)$. We modelled the heat transfer distribution for a one-dimensional segment and discretized it as a linear graph with 21 vertices.

We compared separable and non-separable graph kernels for temporal signals. We used a continuous time-separable kernel as a reference model. We used 50 timestamps for training data and 10 timestamps as test data; we performed ten iterations of backtesting validation with different random seeds and different starting time points. The results are presented in Table 1; we report extrapolation and interpolation MAE with 95% confidence intervals, marking the top re-

Table 1: Interpolation/extrapolation performance for the heat distribution problem over a line (MAE $\times$ 100).

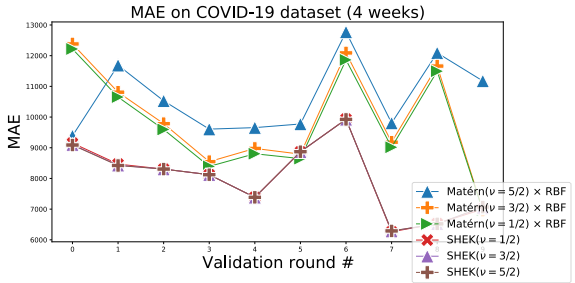
Kernel	MAE_{int}	MAE_{ext}
Laplacian(G) $\times$ RBF(T)	0.42 $\pm$ 3.21	4.75 $\pm$ 0.36
Matérn-1/2(G) $\times$ RBF(T)	**0.05** $\pm$ **0.05**	2.81 $\pm$ 4.12
SHEK(G, T)	0.13 $\pm$ 0.11	**1.29** $\pm$ **0.06**
Exp($\mathbb{R}$) $\times$ Exp(T)	*0.12* $\pm$ *0.12*	*1.02* $\pm$ *0.05*

sults of the graph kernels in bold. We observed that the non-separable stochastic heat kernel outperformed separable kernels on the extrapolation task, but the product graph Matérn kernel and RBF outperformed it on the interpolation task. As expected, exponential kernel over continuous space outperformed graph discretization for extrapolation but did not outperform the product of graph Matérn kernel and RBF over time.

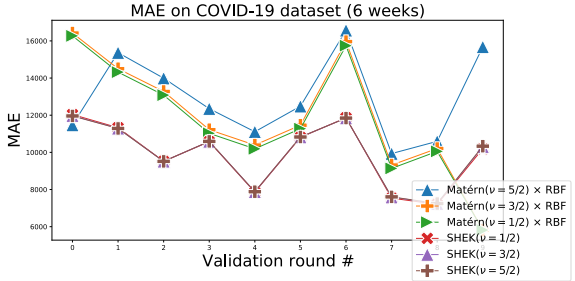
5.2 COVID-19 Distribution Across the US

As a real-world use-case of the proposed method, we considered the prediction of COVID-19 distribution and spread across the USA. We used data about COVID-19 cases and deaths published by The New York Times (2020). We aggregated the number of cases by week for each state. Moreover, we handcrafted a graph that represents the geographical neighbors of the states in the US. Each node of the graph represents a state, and two nodes are connected with an undirected edge if the corresponding states share a common border. The approach can be extended to other graphs that arise from, for example, traffic links or information about the flights. As a modelling target for the extrapolation task, we aimed to predict how many cases there will be in each state during the next two weeks. For numerical stability, we predicted the natural logarithm of this value. We used 33 weeks as training data and estimated the number of cases for the subsequent two weeks. We ran all experiments ten times with a sliding starting point and different random seeds. The goal of the experiment was to show that the proposed kernels can achieve good results within the GP framework. We argue that GPs are particularly useful in this case because of their excellent interpolation properties and the possibility to assess the uncertainties, which is crucial for decision-making. We compared the results with a graph neural network (GNN), for which we used a diffusion convolutional recurrent neural network (DCRNN) layer (Li et al., 2017) followed by a fully connected layer. We used the implementation from the library 'PyTorch Geometric Temporal' (Rozemberczki et al., 2021a). We also deepened the architecture but the results remained largely unaffected. For this reason, we report the results for the simplest architecture.

The results are shown in Table 2. We can see that the models converged to a relatively low MAPE, and SHEK showed the best performance. This shows that the proposed methods are useful for the task we consider. The results on the interpolation task showed better performance for SHEK (DM-test, $p = 0.1$). The results on extrapolation performance are equal for considered separable and non-separable cases. However, SHEK's performance was more stable across validation rounds on the extrapolation task, resulting in two times smaller 95% confidence intervals of generalization error. When the extrapolation period was increased to four and six weeks, SHEK performed consistently better than other kernels: four weeks period MAE of SHEK($\nu = 1/2$) was less than MAE of the product of Matérn(1/2) and RBF (DM-test, $p < 0.01$), as well as on six week period (DM-test, $p < 0.05$).



(a) Four weeks extrapolation.



(b) Six weeks extrapolation.

Figure 3: Visualization of extrapolation evaluation with extended extrapolation periods of four and six weeks on COVID-19 dataset.

Experiment with a flight graph. Additionally, We conducted an experiment to check whether non-spatial graph information can be included in the Covid-19 modeling task. We collected information about flights between the states and used this graph instead of geographical adjacencies. Results showed MAE statistically better for non-separable ($p = 0.05$, DM-test); MAPE showed no statistically significant difference between separable and non-separable kernels in this case.

Table 2: Summary of the interpolation (int) and k-weeks extrapolation (ext-k) results in terms of mean absolute error (MAE) and mean absolute percentage error (MAPE) on COVID-19 and Hungarian chickenpox data sets. Note that the DCRNN model does not support interpolation.

	Chickenpox in Hungary			COVID-19 in the US	
Model	MAE_{int}	$\text{MAE}_{\text{ext-4}}$	$\text{MAE}_{\text{ext-6}}$	MAPE_{int}	$\text{MAPE}_{\text{ext-2}}$
Laplacian(G) × RBF(T)	33.95 ± 1.64	34.25 ± 1.68	46.45 ± 4.93	1.58 ± 0.53	0.76 ± 0.03
Matérn-3/2(G) × RBF(T)	**13.82 ± 0.72**	**26.45 ± 4.05**	32.34 ± 3.07	0.17 ± 0.02	**0.27 ± 0.08**
DCRNN(G,T)	—	27.96 ± 3.65	**30.45 ± 2.19**	—	0.56 ± 0.07
SHEK(G, T) ($\nu = \frac{1}{2}$, $\kappa = 1$)	14.81 ± 0.91	**26.46 ± 3.73**	**30.65 ± 2.51**	**0.16 ± 0.03**	**0.25 ± 0.04**

5.3 Hungary Chickenpox Data Set

We evaluated our methods on a data set of chickenpox cases in Hungary (Rozemberczki et al., 2021b). The data set contains the weekly aggregated numbers of cases of chickenpox in Hungarian counties and Budapest from 2005 until 2015. The spatial relation of counties is represented as a graph with each county as a node (in total 20 nodes) and unweighted edges that indicate adjacency of the counties (in total 61 edges). This data set has a different structure than the COVID-19: the epidemic has strong seasonality and does not have periods of fast exponential growth.

We observed similar results to the COVID-19 case: GP approaches outperformed the GNN approach, SHEK has shown comparable performance with separable kernel on the extrapolation task with more stable performance across the validation rounds. On the long-term (six weeks) extrapolation task, DCRNN and SHEK outperformed other approaches. The separable kernel has shown better performance on the interpolation task than SHEK (DM-test, $p = 0.05$). We repeated the experiment for the longer extrapolation periods: four and six weeks and observed that SHEK performed more stable and consistently better than separable kernels (DM-test, $p = 0.1$ for four weeks and $p = 0.05$ for six weeks). We also experimented with other types of separable kernels and selected Matérn-3/2 plus RBF as a consistently accurate combination on interpolation and extrapolation tasks. We provide full results of the evaluation and point-wise comparison in Appendix E.2.

6 DISCUSSION & CONCLUSION

We considered spatio-temporal problems on graphs. We showed how these can be addressed by product-separable kernels and, then, introduced a framework to derive spatial and spatio-temporal graph kernels based on SPDEs. As concrete instances, we considered the stochastic heat and wave equations on graphs, and solved them to arrive at novel types of graph kernels: the stochastic heat equation kernel (SHEK) and the stochastic wave equation kernel (SWEK). We compared GPs with the proposed kernels against DCRNNs, and demonstrated that our approach outperforms graph neural networks on some machine learning tasks.

Following our method for deriving kernels from SPDEs, this work creates a possibility for direct use of additional prior information in graph problems by creating new flexible kernels on graphs. These methods can be applied to other types of equations resulting in either an analytical solution or an SPDE that can be numerically solved. It may lead to the more frequent use of probabilistic methods on graphs overcoming predictive power issues with the pre-existing kernels.

Our work opens up new avenues of research, such as the solution of other types of equations on graphs or discretization of continuous kernels using graph approximation. Moreover, it gives machine learning tools that can be used for essential problems on graphs, such as epidemic modelling. As a side contribution of our work, we constructed a timely data set of COVID-19 cases over a graph that can be used to evaluate temporal machine learning problems on graphs.

In future work, the scalability of the methods to larger graphs has to be explored. Possible approaches include state space models, numerical solvers for SPDEs, and sparse methods.

Source code and data for the experiments are available at: `https://github.com/AaltoPML/spatiotemporal-graph-kernels`

Acknowledgments

This work was supported by the Academy of Finland Flagship Programme Finnish Center for Artificial Intelligence FCAI, Academy of Finland grants 339730, 341763 and 324345, and UKRI Turing AI World-Leading Researcher Fellowship, EP/W002973/1. We also acknowledge the computational resources provided by the Aalto Science-IT Project from Computer Science IT. We thank Ivan Yashchuk and Ivan Yaroslavtsev for their helpful comments.

References

Akoglu, L., Tong, H., and Koutra, D. (2015). Graph based anomaly detection and description: a survey. *Data Mining and Knowledge Discovery*, 29(3):626–688.

Álvarez, M. A., Rosasco, L., and Lawrence, N. D. (2012). Kernels for vector-valued functions: A review. *Found. Trends Mach. Learn.*, 4(3):195–266.

Belkin, M. and Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396.

Benzi, M., Bertaccini, D., Durastante, F., and Simunec, I. (2020). Non-local network dynamics via fractional graph laplacians. *Journal of Complex Networks*, 8(3).

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer, Berlin, Heidelberg.

Borovitskiy, V., Azangulov, I., Terenin, A., Mostowsky, P., Deisenroth, M., and Durrande, N. (2021). Matérn Gaussian processes on graphs. In *International Conference on Artificial Intelligence and Statistics*, pages 2593–2601. PMLR.

Borovitskiy, V., Terenin, A., Mostowsky, P., and Deisenroth (he/him), M. (2020). Matérn Gaussian processes on Riemannian manifolds. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12426–12437. Curran Associates, Inc.

Burrough, P. A., McDonnell, R., McDonnell, R. A., and Lloyd, C. D. (2015). *Principles of Geographical Information Systems*. Oxford University Press.

Campbell, W. M., Dagli, C. K., and Weinstein, C. J. (2013). Social network analysis with content and graphs. *Lincoln Laboratory Journal*, 20(1):61–81.

Carrington, P. J., Scott, J., and Wasserman, S. (2005). *Models and Methods in Social Network Analysis*, volume 28. Cambridge University Press.

Chami, I., Abu-El-Haija, S., Perozzi, B., Ré, C., and Murphy, K. (2020). Machine learning on graphs: A model and comprehensive taxonomy. *arXiv preprint arXiv:2005.03675*.

Chung, F. R. and Graham, F. C. (1997). *Spectral Graph Theory*. American Mathematical Soc.

Diebold, F. X. and Mariano, R. S. (2002). Comparing predictive accuracy. *Journal of Business & economic statistics*, 20(1):134–144.

Duvenaud, D. (2014). *Automatic Model Construction with Gaussian Processes*. PhD thesis, University of Cambridge.

Feischl, M., Kuo, F. Y., and Sloan, I. H. (2018). Fast random field generation with H-matrices. *Numerische Mathematik*, 140(3):639–676.

Gardiner, C. W. (2004). *Handbook of Stochastic Methods for Physics, Chemistry and the Natural Sciences*, volume 13 of *Springer Series in Synergetics*. Springer-Verlag, Berlin, third edition.

Grady, L. J. and Polimeni, J. R. (2010). *Discrete calculus: Applied Analysis on Graphs for Computational Science*. Springer Science & Business Media.

Graham, I. G., Kuo, F. Y., Nuyens, D., Scheichl, R., and Sloan, I. H. (2018). Analysis of circulant embedding methods for sampling stationary random fields. *SIAM Journal on Numerical Analysis*, 56(3):1871–1895.

Higham, D. J., Mao, X., and Stuart, A. M. (2002). Strong convergence of euler-type methods for nonlinear stochastic differential equations. *SIAM Journal on Numerical Analysis*, 40(3):1041–1063.

Hooten, M. B., Johnson, D. S., McClintock, B. T., and Morales, J. M. (2017). *Animal Movement: Statistical Models for Telemetry Data*. CRC press, Boca Raton, Florida.

Jiang, W. and Luo, J. (2021). Graph neural network for traffic forecasting: A survey. *arXiv preprint arXiv:2101.11174*.

Keeling, M. J. and Eames, K. T. (2005). Networks and epidemic models. *Journal of the Royal Society Interface*, 2(4):295–307.

Li, Y., Yu, R., Shahabi, C., and Liu, Y. (2017). Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*.

Lindgren, F., Rue, H., et al. (2015). Bayesian spatial modelling with R-INLA. *Journal of Statistical Software*, 63(19):1–25.

Lindgren, F., Rue, H., and Lindström, J. (2011). 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.

Liu, D. C. and Nocedal, J. (1989). On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1):503–528.

Liu, H., Ong, Y.-S., Shen, X., and Cai, J. (2020). When Gaussian process meets big data: A review of scalable GPs. *IEEE Transactions on Neural Networks and Learning Systems*, 31:4405–4423.

Lovász, L. (1993). Random walks on graphs. *Combinatorics, Paul Erdős is Eighty*, 2(1–46):4.

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

Moore, E. H. (1920). On the reciprocal of the general algebraic matrix. *Bulletin of American Mathematical Society*, 26:394–395.

Moraga, P., Cramb, S. M., Mengersen, K. L., and Pagano, M. (2017). A geostatistical model for combined analysis of point-level and area-level data using INLA and SPDE. *Spatial Statistics*, 21:27–41.

Oksendal, B. (2013). *Stochastic Differential Equations: an Introduction with Applications*. Springer Science & Business Media.

Rasmussen, C. and Williams, C. (2006). *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, USA.

Romero, D., Ioannidis, V. N., and Giannakis, G. B. (2017). Kernel-based reconstruction of space-time functions on dynamic graphs. *IEEE Journal of Selected Topics in Signal Processing*, 11(6):856–869.

Romero, D., Ma, M., and Giannakis, G. B. (2016). Kernel-based reconstruction of graph signals. *IEEE Transactions on Signal Processing*, 65(3):764–778.

Rozemberczki, B., Scherer, P., He, Y., Panagopoulos, G., Riedel, A., Astefanoaei, M., Kiss, O., Beres, F., Lopez, G., Collignon, N., and Sarkar, R. (2021a). PyTorch Geometric Temporal: Spatiotemporal signal processing with neural machine learning models. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*.

Rozemberczki, B., Scherer, P., Kiss, O., Sarkar, R., and Ferenci, T. (2021b). Chickenpox cases in hungary: a benchmark dataset for spatiotemporal signal processing with graph neural networks. *arXiv preprint arXiv:2102.08100*.

Ruge, J. W. and Stüben, K. (1987). Algebraic multigrid. In *Multigrid Methods*, pages 73–130. SIAM.

Sanz-Alonso, D. and Yang, R. (2020). The SPDE approach to Matérn fields: Graph representations. *arXiv preprint arXiv:2004.08000*.

Särkkä, S. (2011). Linear operators and stochastic partial differential equations in gaussian process regression. In *Artificial Neural Networks and Machine Learning – ICANN 2011*, pages 151–158, Berlin, Heidelberg. Springer Berlin Heidelberg.

Särkkä, S. and Solin, A. (2019). *Applied Stochastic Differential Equations*. Cambridge University Press.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. (2008). The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80.

Schottl, A. (1997). Stochastic partial differential equations: A modeling, white noise functional approach. *Journal of the American Statistical Association*, 92(440):1650–1652.

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

Solin, A. (2016). *Stochastic Differential Equation Methods for Spatio-Temporal Gaussian Process Regression*. PhD thesis, Aalto University.

The New York Times (2020). Coronavirus (Covid-19) data in the United States. `https://github.com/nytimes/covid-19-data` (accessed 2021-01-04).

Tsuda, K. (2011). Graph classification methods in chemoinformatics. In *Handbook of Statistical Bioinformatics*, pages 335–351. Springer.

Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416.

Wardetzky, M. (2020). Discrete Laplace operators. *An Excursion Through Discrete Differential Geometry: AMS Short Course, Discrete Differential Geometry, January 8-9, 2018, San Diego, California*, 76:1.

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

Yan, S., Xu, D., Zhang, B., Zhang, H., Yang, Q., and Lin, S. (2007). Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1):40–51.

Yu, B., Yin, H., and Zhu, Z. (2018). Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3634–3640.

Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., Deng, M., and Li, H. (2019). T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, 21(9):3848–3858.

Supplementary Material for Non-separable Spatio-temporal Graph Kernels via SPDEs

A Preliminaries

In our proofs, we rely on the following theorem.

Theorem A.1 (Itô isometry, Oksendal (2013))**.**

$$\mathbb{E}\left[\left(\int_S^T f(t,w)\,\mathrm{d}W_t\right)^2\right] = \mathbb{E}\left[\int_S^T f^2(t,w)\,\mathrm{d}t\right], \tag{A.1}$$

for all $f \in \mathcal{V}(S,T)$.

Corollary A.1.1.

$$\mathbb{E}\left[\left(\int_0^t X_t\,\mathrm{d}W_t\right)\left(\int_0^t Y_t\,\mathrm{d}W_t\right)\right] = \mathbb{E}\left[\int_0^t X_t Y_t\,\mathrm{d}t\right], \tag{A.2}$$

where X_t *and* Y_t *are stochastic processes adapted to natural filtration. This property will be used actively in derivations of covariances from SDEs.*

Lemma A.2 (Integration of the matrix exponential)**.**

$$\int_0^T e^{\boldsymbol{A}t}\,\mathrm{d}t = \left(e^{\boldsymbol{A}T} - \boldsymbol{I}\right)\boldsymbol{A}^{-1}, \tag{A.3}$$

where $\boldsymbol{A}$ *is a nonsingular matrix.*

B Graph Kernels

In this section, we introduce spatial kernels on graphs derived by the SPDE framework.

Proposition B.1 (Laplace's equation in kernel form)**.** *Consider a signal (process)* $\boldsymbol{v} \in \mathbb{R}^{|V|}$ *on a graph* G*, where* $\boldsymbol{L}$ *is the Laplacian of* G *and the signal is described with the Laplace SPDE on the graph:* $-\boldsymbol{L}\boldsymbol{v} = \boldsymbol{w}$*. Then the solution* $\boldsymbol{v}$ *can be described as a Gaussian process:*

$$\boldsymbol{v} \sim \mathcal{N}\left(\boldsymbol{0}, (\boldsymbol{L}^*\boldsymbol{L})^+\right). \tag{B.1}$$

Proof. By writing out

$$\boldsymbol{v} = -\boldsymbol{L}^+\boldsymbol{w}, \tag{B.2}$$

and noting that the covariance matrix of white noise $\boldsymbol{w}$ is $\boldsymbol{I}$, the covariance of the random variable $\boldsymbol{u}$ is

$$\mathrm{Cov}\left[-\boldsymbol{L}^+\boldsymbol{w}\right] = \boldsymbol{L}^+\mathrm{Cov}\left[\boldsymbol{w}\right]\boldsymbol{L}^{+\top} = \left(\boldsymbol{L}^\top\boldsymbol{L}\right)^+. \tag{B.3}$$

□

Proposition B.2 (Matérn kernels on graphs)**.** *Matérn kernels on graphs are described with the equation*

$$\left(\frac{2\nu}{\kappa^2}\boldsymbol{I} + \boldsymbol{L}\right)^{\frac{\nu}{2}}\boldsymbol{v} = \boldsymbol{w}, \tag{B.4}$$

and have the form

$$\boldsymbol{K}_{XX} = \left[\left(\left(\frac{2\nu}{\kappa^2}\boldsymbol{I} + \boldsymbol{L}\right)^{-\frac{\nu}{2}}\right)^\top \left(\frac{2\nu}{\kappa^2}\boldsymbol{I} + \boldsymbol{L}\right)^{-\frac{\nu}{2}}\right]^{-1}. \tag{B.5}$$

For the self-adjoint Laplacian $\boldsymbol{L}$:

$$\boldsymbol{K}_{XX} = \left(\frac{2\nu}{\kappa^2}\boldsymbol{I} + \boldsymbol{L}\right)^{\nu}. \tag{B.6}$$

Thus, the solution of Eq. (B.4) *is*

$$\boldsymbol{v} \sim \mathcal{N}\Big(\boldsymbol{0}, \boldsymbol{K}_{XX}\Big). \tag{B.7}$$

Proof. Let $\boldsymbol{A} = \left(\frac{2\nu}{\kappa^2}\boldsymbol{I} + \boldsymbol{L}\right)^{\frac{\nu}{2}}$, then $\boldsymbol{v} = \boldsymbol{A}^{-1}\boldsymbol{w}$. Now, we can derive a covariance matrix, analogously as in Proposition B.1. □

C Stochastic Heat and Wave Equations on Graphs

C.1 Heat Equation

Proposition C.1 (Heat equation)**.** *The heat equation on graphs is defined by the second order matrix differential equation*

$$\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} = -c\,\boldsymbol{L}\boldsymbol{u}, \tag{C.1}$$

and the solution of this equation is:

$$\boldsymbol{K}_H = e^{-c\boldsymbol{L}t}\boldsymbol{u}(0). \tag{C.2}$$

Proof. The solution of the differential equation is similar to the homogeneous initial value problem in the continuous case and can be written in terms of a matrix exponential:

$$\boldsymbol{u}(t) = e^{-c\boldsymbol{L}t}\boldsymbol{u}(0). \tag{C.3}$$

□

For stochastic heat and wave equations, we will be using the pseudo-differential operator $\widetilde{\boldsymbol{L}}$ instead of the graph Laplacian because this operator is more flexible than L (for $\nu = 1$ and $\kappa \to \infty$, it coincides with L), and because it possesses non-local properties as was shown by Benzi et al. (2020).

Proposition C.2 (Stochastic heat equation on graphs)**.** *The stochastic heat equation can be defined on a graph by adding spatio-temporal white noise, or for convenient integration, as the* formal *derivative of Wiener process $\dot{\boldsymbol{W}}_t$:*

$$\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} = -c\widetilde{\boldsymbol{L}}\boldsymbol{u} + \sigma\dot{\boldsymbol{W}}_t. \tag{C.4}$$

The solution can be defined as a Gaussian process:

$$\begin{aligned}
\boldsymbol{u}(t) &\sim \mathcal{GP}(\boldsymbol{\mu}(t), \mathrm{Cov}[\boldsymbol{u}(s), \boldsymbol{u}(t)]), \quad \textit{with} \\
\boldsymbol{\mu}(t) &= e^{-c\widetilde{\boldsymbol{L}}t}\boldsymbol{u}(0), \\
\mathrm{Cov}[\boldsymbol{u}(t), \boldsymbol{u}(s)] &= \frac{\sigma^2}{c} e^{-c\widetilde{\boldsymbol{L}}t - c\widetilde{\boldsymbol{L}}^\top s}\left(e^{c(\widetilde{\boldsymbol{L}}+\widetilde{\boldsymbol{L}}^\top)\min(t,s)} - \boldsymbol{I}\right)(\widetilde{\boldsymbol{L}} + \widetilde{\boldsymbol{L}}^\top)^{-1}.
\end{aligned} \tag{C.5}$$

Or, when the matrix $\widetilde{\boldsymbol{L}}$ is self-adjoint (the graph is undirected), as

$$\begin{aligned}
\boldsymbol{\mu}(t) &= e^{-c\widetilde{\boldsymbol{L}}t}\boldsymbol{u}(0), \\
\mathrm{Cov}[\boldsymbol{u}(t), \boldsymbol{u}(s)] &= \frac{\sigma^2}{2c}\left(e^{-c\widetilde{\boldsymbol{L}}|t-s|} - e^{-c\widetilde{\boldsymbol{L}}(t+s)}\right)\widetilde{\boldsymbol{L}}^{-1}.
\end{aligned} \tag{C.6}$$

Proof. Let $\boldsymbol{\Gamma} = c\widetilde{\boldsymbol{L}}$. Equation (C.4) is a matrix differential equation and a solution can be written in the form:

$$\boldsymbol{u}(t) = e^{-\boldsymbol{\Gamma} t}\boldsymbol{u}(0) + \sigma e^{-\boldsymbol{\Gamma} t}\int_0^t e^{\boldsymbol{\Gamma} s}\,\mathrm{d}\boldsymbol{W}_s. \tag{C.7}$$

The solution in Eq. (C.7) can be expressed as a GP. We give the corresponding mean and covariance as follows.

$$\mathbb{E}[\boldsymbol{u}(t)] = e^{-\boldsymbol{\Gamma} t}\boldsymbol{u}(0). \tag{C.8}$$

Using Itô isometry, we can derive the covariance:

$$\begin{aligned}
\mathrm{Cov}\left[\boldsymbol{u}(s), \boldsymbol{u}(t)\right] &= \mathbb{E}\left[\left(\boldsymbol{u}(s) - \mathbb{E}[\boldsymbol{u}(s)]\right)\left(\boldsymbol{u}(t) - \mathbb{E}[\boldsymbol{u}(t)]\right)^\top\right] \\
&= \mathbb{E}\left[\left(\sigma e^{-\boldsymbol{\Gamma} s}\int_0^s e^{\boldsymbol{\Gamma}\xi} d\boldsymbol{W}_\xi\right)\left(\sigma e^{-\boldsymbol{\Gamma}^\top t}\int_0^t e^{\boldsymbol{\Gamma}^\top \xi} d\boldsymbol{W}_\xi\right)\right] \\
&= \sigma^2 e^{-\boldsymbol{\Gamma} t - \boldsymbol{\Gamma}^T s}\,\mathbb{E}\left[\left(\int_0^s e^{\boldsymbol{\Gamma}\xi} d\boldsymbol{W}_\xi\right)\left(\int_0^t e^{\boldsymbol{\Gamma}^T \xi} d\boldsymbol{W}_\xi\right)\right] \\
&= \sigma^2 e^{-\boldsymbol{\Gamma} t - \boldsymbol{\Gamma}^T s}\left(e^{\left(\boldsymbol{\Gamma}+\boldsymbol{\Gamma}^\top\right)\min(t,s)} - \boldsymbol{I}\right)\left(\boldsymbol{\Gamma} + \boldsymbol{\Gamma}^\top\right)^{-1}.
\end{aligned} \tag{C.9}$$

Or, in the case of self-adjoint $\widetilde{\boldsymbol{L}}$ the expression can be simplified:

$$\frac{\sigma^2}{2}\left(e^{-\boldsymbol{\Gamma}|t-s|} - e^{-\boldsymbol{\Gamma}(t+s)}\right)\boldsymbol{\Gamma}^{-1}. \tag{C.10}$$

□

Proposition C.3 (Stochastic heat equation on undirected graphs with matrix-scaled white noise)**.** *Let us consider the same equation as in the previous theorem but with $\boldsymbol{\Sigma}$-scaled white noise on undirected connected graphs:*

$$\mathrm{d}\boldsymbol{u} = -c\widetilde{\boldsymbol{L}}\boldsymbol{u}\,\mathrm{d}t + \boldsymbol{\Sigma}\,\mathrm{d}\boldsymbol{W_t}. \tag{C.11}$$

The covariance can be derived as follows. Consider $\boldsymbol{\Gamma} = c\widetilde{\boldsymbol{L}}$ and a diagonalization of $\boldsymbol{\Gamma}$:

$$\boldsymbol{P}\widetilde{\boldsymbol{L}}\boldsymbol{P}^* = \boldsymbol{P}\widetilde{\boldsymbol{L}}^T\boldsymbol{P}^+ = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n). \tag{C.12}$$

Then, for $t \geq s$:

$$\mathrm{Cov}\left[\boldsymbol{u}(t), \boldsymbol{u}(s)\right] = \boldsymbol{P}^*\boldsymbol{C}(t,s)\boldsymbol{P}, \tag{C.13}$$

where $\boldsymbol{C}(t,s)$ is defined as:

$$\boldsymbol{C}(t,s)_{i,j} = \frac{1}{c}\frac{(\boldsymbol{P}\boldsymbol{\Sigma}\boldsymbol{\Sigma}^\top\boldsymbol{P}^*)_{i,j}}{\lambda_i + \lambda_j}(\exp(-\lambda_i c|t-s|) - \exp(-c(\lambda_i t + \lambda_j s))). \tag{C.14}$$

Proof. Let us write covariance between $\boldsymbol{u}(t)$ and $\boldsymbol{u}(s)$ and apply Itô isometry

$$\begin{aligned}
\mathrm{Cov}[\boldsymbol{u}(t), \boldsymbol{u}^\top(s)] &= \mathbb{E}\left[\left(\boldsymbol{u}(t) - \mathbb{E}[\boldsymbol{u}(t)]\right)\left(\boldsymbol{u}(s) - \mathbb{E}[\boldsymbol{u}(s)\right)^\top\right] \\
&= \int_0^{\min(t,s)} \exp(-c\widetilde{\boldsymbol{L}}(t-\xi))\boldsymbol{\Sigma}\boldsymbol{\Sigma}^\top \exp(-c\widetilde{\boldsymbol{L}}(s-\xi))\mathrm{d}\xi.
\end{aligned} \tag{C.15}$$

Consider a diagonalization $\boldsymbol{P}\widetilde{\boldsymbol{L}}\boldsymbol{P}^* = \boldsymbol{L}_d = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$. Then the covariance can be re-written as

$$\mathrm{Cov}[\boldsymbol{u}(t), \boldsymbol{u}^\top(s)] = \boldsymbol{P}^*\underbrace{\left(\int_0^{\min(t,s)} \exp(-c\boldsymbol{L}_d(t-\xi))\boldsymbol{P}\boldsymbol{\Sigma}\boldsymbol{\Sigma}^\top\boldsymbol{P}^* \exp(-c\boldsymbol{L}_d(s-\xi))\mathrm{d}\xi\right)}_{\boldsymbol{C}(t,s)}\boldsymbol{P}. \tag{C.16}$$

Because the matrix $\boldsymbol{L}_d$ is diagonal, we can write the $\boldsymbol{C}(t,s)_{ij}$ (for $t \geq s$):

$$\begin{aligned}\boldsymbol{C}(t,s)_{ij} &= (\boldsymbol{P}\boldsymbol{\Sigma}\boldsymbol{\Sigma}^\top\boldsymbol{P}^*)_{ij}\int_0^s \exp(-c\lambda_i t - c\lambda_j s + c\xi(\lambda_i+\lambda_j))\,\mathrm{d}\xi \\ &= \frac{1}{c}\frac{\boldsymbol{P}\boldsymbol{\Sigma}\boldsymbol{\Sigma}^\top\boldsymbol{P}^*}{\lambda_i+\lambda_j}\left(\exp(c\lambda_i|t-s|) - \exp(-c\lambda_i t - c\lambda_j s)\right). \end{aligned} \tag{C.17}$$

□

C.2 Wave Equation

Proposition C.4 (Wave equation on an undirected graph)**.** *The wave kernel on undirected graphs is defined by the second order matrix differential equation*

$$\frac{\mathrm{d}^2\boldsymbol{u}}{\mathrm{d}t^2} = -c^2\boldsymbol{L}\boldsymbol{u}, \tag{C.18}$$

and the solution of this equation has a form:

$$\boldsymbol{u}(t) = \frac{1}{c}\sqrt{\boldsymbol{L}^+}\sin\left(c\sqrt{\boldsymbol{L}}t\right)\boldsymbol{P}\dot{\boldsymbol{u}}(0) + \cos\left(c\sqrt{\boldsymbol{L}}t\right)\boldsymbol{P}\boldsymbol{u}(0). \tag{C.19}$$

Proof. $\boldsymbol{L}$ is a symmetric matrix and, consequently, diagonalizable:

$$\boldsymbol{L} = \boldsymbol{P}^{-1}\boldsymbol{L}_d\boldsymbol{P}, \tag{C.20}$$

Then the equation will take a form:

$$\ddot{\boldsymbol{u}} + c^2\boldsymbol{P}^{-1}\boldsymbol{L}_d\boldsymbol{P}\boldsymbol{u} = \boldsymbol{0} \Leftrightarrow \boldsymbol{P}\ddot{\boldsymbol{u}} + c^2\boldsymbol{L}_d\boldsymbol{P}\boldsymbol{u} = \boldsymbol{0}. \tag{C.21}$$

Replacing the variable $\boldsymbol{y} \coloneqq \boldsymbol{P}\boldsymbol{u}$, we will got the following equation:

$$\ddot{\boldsymbol{y}} + c^2\boldsymbol{L}_d\boldsymbol{y} = \boldsymbol{0}. \tag{C.22}$$

This is equivalent to n independent scalar ODEs:

$$\ddot{y}_k + c^2\lambda_k y_k = 0. \tag{C.23}$$

The solution of each of these equation is:

$$y_k = c_1\sin\left(c\sqrt{\lambda_k}t\right) + c_2\cos\left(c\sqrt{\lambda_k}t\right). \tag{C.24}$$

Let the initial conditions of the system be given as $\boldsymbol{y}(0), \dot{\boldsymbol{y}}(0)$. Then, for $\lambda_k \neq 0$ c_1 and c_2 can be expressed by

$$\begin{cases} c_1 = \frac{1}{c\sqrt{\lambda_k}}\dot{y}_k(0) \\ c_2 = \quad y_k(0) \end{cases}$$

and $c_1 = c_2 = 0$ for $\lambda_k = 0$.

By doing the reverse substitution, we get the result for u:

$$\boldsymbol{u}(t) = \frac{1}{c}\sqrt{\boldsymbol{L}^+}\sin\left(c\sqrt{\boldsymbol{L}}t\right)\boldsymbol{P}\dot{\boldsymbol{u}}(0) + \cos\left(c\sqrt{\boldsymbol{L}}t\right)\boldsymbol{P}\boldsymbol{u}(0). \tag{C.25}$$

□

Proposition C.5. *The **Stochastic wave equation kernel (SWEK)** on undirected graphs is defined by second order matrix differential equation*

$$\frac{\mathrm{d}^2\boldsymbol{u}}{\mathrm{d}t^2} = -c^2\widetilde{\boldsymbol{L}}\boldsymbol{u} + \sigma\dot{\boldsymbol{W}}_t \tag{C.26}$$

and the solution to this equation can be expressed as a Gaussian process:

$$\boldsymbol{u}(t) \sim \mathcal{GP}(\boldsymbol{\mu}, \mathrm{Cov}[\boldsymbol{u}(s), \boldsymbol{u}(t)] \tag{C.27}$$

$$\boldsymbol{\mu} = \frac{1}{c}\widetilde{\boldsymbol{L}}^{-\frac{1}{2}} \sin\left(c\sqrt{\widetilde{\boldsymbol{L}}}t\right)\boldsymbol{P}\dot{\boldsymbol{u}}(0) + \cos\left(c\sqrt{\widetilde{\boldsymbol{L}}}t\right)\boldsymbol{P}\boldsymbol{u}(0) \tag{C.28}$$

$$\mathrm{Cov}[\boldsymbol{u}(s), \boldsymbol{u}(t)] = \sigma^2\boldsymbol{\Theta}^{-2}\left(\cos(\boldsymbol{\Theta}(t-s))\,\min(t,s) - \frac{1}{2}\cos(\boldsymbol{\Theta}\max(t,s))\sin(\boldsymbol{\Theta}\min(t,s))\boldsymbol{\Theta}^{-1}\right). \tag{C.29}$$

Here, $\boldsymbol{\Theta} = c\sqrt{\widetilde{\boldsymbol{L}}}$.

Proof. The homogeneous solution for wave equation is given in Theorem C.4:

$$\boldsymbol{u}(t) = \widetilde{\boldsymbol{L}}^{-\frac{1}{2}} \sin\left(c\sqrt{\widetilde{\boldsymbol{L}}}t\right)c_1 + \cos\left(c\sqrt{\widetilde{\boldsymbol{L}}}t\right)c_2.$$

Let us define $\boldsymbol{\Theta} = c\sqrt{\widetilde{\boldsymbol{L}}}$ for convenience. In order to solve the given SPDE we need to find the inhomogeneous part. It can be found by variation of the parameters. The Wronskian of the basis functions (here denoted as $\mathbf{Wr}$ to distinguish it from the Wiener process) is:

$$\mathbf{Wr}(v_1, v_2)(t) = \begin{vmatrix} \cos(\boldsymbol{\Theta}t) & \sin(\boldsymbol{\Theta}t) \\ \frac{\mathrm{d}}{\mathrm{d}t}\cos(\boldsymbol{\Theta}t) & \frac{\mathrm{d}}{\mathrm{d}t}\sin(\boldsymbol{\Theta}t) \end{vmatrix} = \boldsymbol{\Theta}, \tag{C.30}$$

for $\boldsymbol{v}_1 = \cos(\boldsymbol{\Theta}t)$ and $\boldsymbol{v}_2 = \sin(\boldsymbol{\Theta}t)$. The particular solution will have the form:

$$\boldsymbol{u}(t) = \mathbf{Wr}(\boldsymbol{v}_1, \boldsymbol{v}_2)^{-1}\left(-\boldsymbol{v}_1(t)\int \boldsymbol{v}_2(t)\,\mathrm{d}\boldsymbol{W}_t + \boldsymbol{v}_2(t)\int \boldsymbol{v}_1(t)\,\mathrm{d}\boldsymbol{W}_t\right) \tag{C.31}$$

$$= \boldsymbol{\Theta}^{-1}\left(-\cos(\boldsymbol{\Theta}t)\int_0^t \sin(\boldsymbol{\Theta}\xi)\,\mathrm{d}\boldsymbol{W}_\xi + \sin(\boldsymbol{\Theta}t)\int_0^t cos(\boldsymbol{\Theta}\xi)\,\mathrm{d}\boldsymbol{W}_\xi\right) \tag{C.32}$$

Then the solutions for the stochastic wave equation on graphs are

$$\boldsymbol{u}(t) = \boldsymbol{v}_1(t) + \boldsymbol{v}_2(t) + \boldsymbol{u}_0(t) \tag{C.33}$$

$$= \cos(\boldsymbol{\Theta}t)c_1 + \sin(\boldsymbol{\Theta}t)c_2 + \boldsymbol{\Theta}^{-1}\left(-\cos(\boldsymbol{\Theta}t)\int_0^t \sin(\boldsymbol{\Theta}\xi)\,\mathrm{d}\boldsymbol{W}_\xi + \sin(\boldsymbol{\Theta}t)\int_0^t \cos(\boldsymbol{\Theta}\xi)\,\mathrm{d}\boldsymbol{W}_\xi\right). \tag{C.34}$$

Assuming that $\boldsymbol{u}(t) \sim \mathcal{N}(\boldsymbol{\mu}, Cov[\boldsymbol{u}(s), \boldsymbol{u}(t)]$, let us calculate mean and covariance from Equation (C.31):

$$\mathbb{E}[\boldsymbol{u}(t)] = c_1\cos(\boldsymbol{\Theta}t) + c_2\sin(\boldsymbol{\Theta}t), \tag{C.35}$$

$$\begin{aligned}\mathrm{Cov}[\boldsymbol{u}(s), \boldsymbol{u}(t)] &= \mathbb{E}\left[\left(\boldsymbol{u}(s) - \mathbb{E}[\boldsymbol{u}(s)]\right)\left(\boldsymbol{u}(t) - \mathbb{E}[\boldsymbol{u}(t)]\right)^\top\right] \\ &= \sigma^2\boldsymbol{\Theta}^{-2}\left(-\cos(\boldsymbol{\Theta}t)\int_0^t \sin(\boldsymbol{\Theta}\xi)\,\mathrm{d}\boldsymbol{W}_\xi + \sin(\boldsymbol{\Theta}t)\int_0^t \cos(\boldsymbol{\Theta}\xi)\,\mathrm{d}\boldsymbol{W}_\xi\right) \\ &\quad \times \left(-\cos(\boldsymbol{\Theta}s)\int_0^s \sin(\boldsymbol{\Theta}\xi)\,\mathrm{d}\boldsymbol{W}_\xi + \sin(\boldsymbol{\Theta}s)\int_0^s \cos(\boldsymbol{\Theta}\xi)\,\mathrm{d}\boldsymbol{W}_\xi\right)^\top \\ &= \sigma^2\boldsymbol{\Theta}^{-2}\boldsymbol{C}(s,t),\end{aligned} \tag{C.36}$$

where $\boldsymbol{C}(s,t)$ is defined as

$$\boldsymbol{C}(s,t) \triangleq \left(-\cos(\boldsymbol{\Theta}t)\int_0^t \sin(\boldsymbol{\Theta}\xi)\,\mathrm{d}\boldsymbol{W}_\xi + \sin(\boldsymbol{\Theta}t)\int_0^t \cos(\boldsymbol{\Theta}\xi)\,\mathrm{d}\boldsymbol{W}_\xi \right) \times \left(-\cos(\boldsymbol{\Theta}s)\int_0^s \sin(\boldsymbol{\Theta}\xi)\,\mathrm{d}\boldsymbol{W}_\xi + \sin(\boldsymbol{\Theta}s)\int_0^s \cos(\boldsymbol{\Theta}\xi)\,\mathrm{d}\boldsymbol{W}_\xi \right)^\top. \quad \text{(C.37)}$$

Opening the brackets in $\boldsymbol{C}(s,t)$ and using Itô isometry, we will get a sum of the four following expressions:

1.
$$\cos(\boldsymbol{\Theta}t)\cos(\boldsymbol{\Theta}s)\int_0^{\min(t,s)} \sin^2(\boldsymbol{\Theta}\xi)\,\mathrm{d}\xi = \cos(\boldsymbol{\Theta}t)\cos(\boldsymbol{\Theta}s)\left(\frac{\xi}{2}\boldsymbol{I} - \frac{\sin(2\boldsymbol{\Theta}\xi)}{4}\boldsymbol{\Theta}^{-1}\right)\Bigg|_0^{\min(t,s)}, \quad \text{(C.38)}$$

2.
$$\sin(\boldsymbol{\Theta}t)\sin(\boldsymbol{\Theta}s)\int_0^{\min(t,s)} \cos^2(\boldsymbol{\Theta}\xi)\mathrm{d}\xi = \sin(\boldsymbol{\Theta}t)\sin(\boldsymbol{\Theta}s)\left(\frac{\xi}{2}\boldsymbol{I} + \frac{\sin(2\boldsymbol{\Theta}\xi)}{4}\boldsymbol{\Theta}^{-1}\right)\Bigg|_0^{min(t,s)}, \quad \text{(C.39)}$$

3.
$$-\cos(\boldsymbol{\Theta}t)\sin(\boldsymbol{\Theta}s)\int_0^{\min(t,s)} \cos(\boldsymbol{\Theta}\xi)\sin(\boldsymbol{\Theta}\xi)d\xi = -\cos(\boldsymbol{\Theta}t)\sin(\boldsymbol{\Theta}s)\left(-\frac{\cos^2(\boldsymbol{\Theta}\xi)}{2}\boldsymbol{\Theta}^{-1}\right)\Bigg|_0^{min(t,s)}, \quad \text{(C.40)}$$

4.
$$-\sin(\boldsymbol{\Theta}t)\cos(\boldsymbol{\Theta}s)\int_0^{min(t,s)} \cos(\boldsymbol{\Theta}\xi)\sin(\boldsymbol{\Theta}\xi)d\xi = -\sin(\boldsymbol{\Theta}t)\cos(\boldsymbol{\Theta}s)\left(\frac{-\cos^2(\boldsymbol{\Theta}\xi)}{2}\boldsymbol{\Theta}^{-1}\right)\Bigg|_0^{min(t,s)}. \quad \text{(C.41)}$$

Then, the covariance is

$$\begin{aligned}\boldsymbol{C}(s,t) &= \frac{\min(t,s)}{2}\left(\cos(\boldsymbol{\Theta}t)\cos(\boldsymbol{\Theta}s) + \sin(\boldsymbol{\Theta}t)\sin(\boldsymbol{\Theta}s)\right) \\ &\quad + \Bigg(-\cos(\boldsymbol{\Theta}t)\cos(\boldsymbol{\Theta}s)\frac{\sin(2\boldsymbol{\Theta}\min(t,s))}{4} + \sin(\boldsymbol{\Theta}t)\sin(\boldsymbol{\Theta}s)\frac{\sin(2\boldsymbol{\Theta}\min(t,s))}{4} \\ &\qquad + \cos(\boldsymbol{\Theta}t)\sin(\boldsymbol{\Theta}s)\left(\frac{\cos^2(\boldsymbol{\Theta}\min(t,s))}{2} - \frac{1}{2}\right) + \sin(\boldsymbol{\Theta}t)\cos(\boldsymbol{\Theta}s)\left(\frac{\cos^2(\boldsymbol{\Theta}\min(t,s))}{2} - \frac{1}{2}\right)\Bigg)\boldsymbol{\Theta^{-1}} \\ &= \cos(\boldsymbol{\Theta}(t-s))\ \min(t,s) - \frac{1}{2}\cos(\boldsymbol{\Theta}\max(t,s))\sin(\boldsymbol{\Theta}\min(t,s))\boldsymbol{\Theta}^{-1}. \end{aligned} \quad \text{(C.42)}$$

□

D Kernel Visualizations

In Figure S1, we can observe behavior expected for heat and wave processes that start from the third node on a line graph. We observed similar visualizations for other types of graphs.

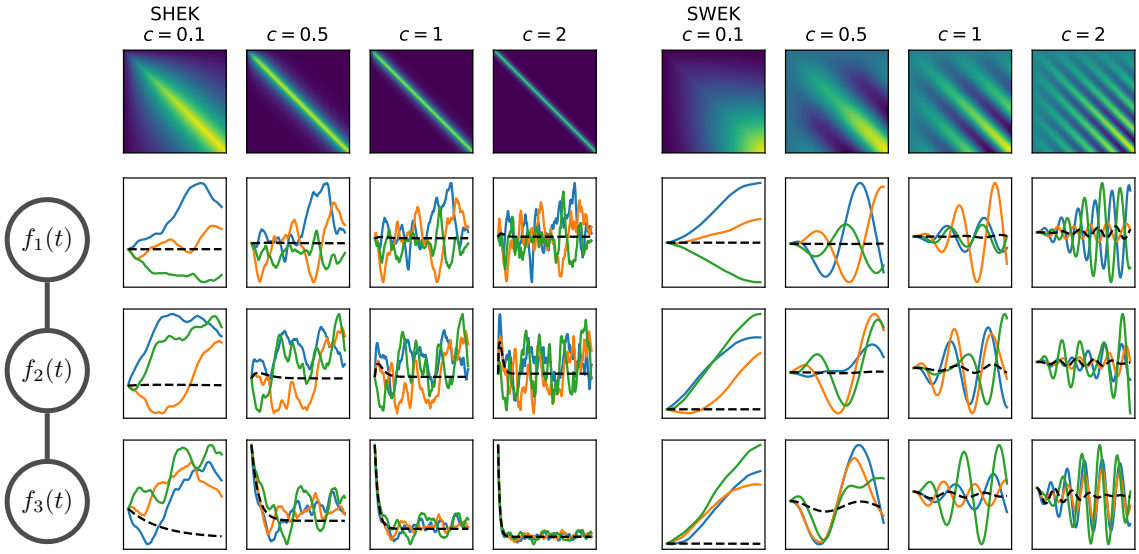


Figure S1: Temporal visualizations of SHEK (heat, left) and SWEK (wave, right) on a linear three-node graph. The first row shows the temporal part of the covariance matrix (summed over the graph vertices at each timepoint). The following three rows show mean (black) and samples (colored lines) as a function of time at each of the nodes, conditioned on $y(t = 0) = (0, 0, 10)$, for different values of the hyperparameter c.

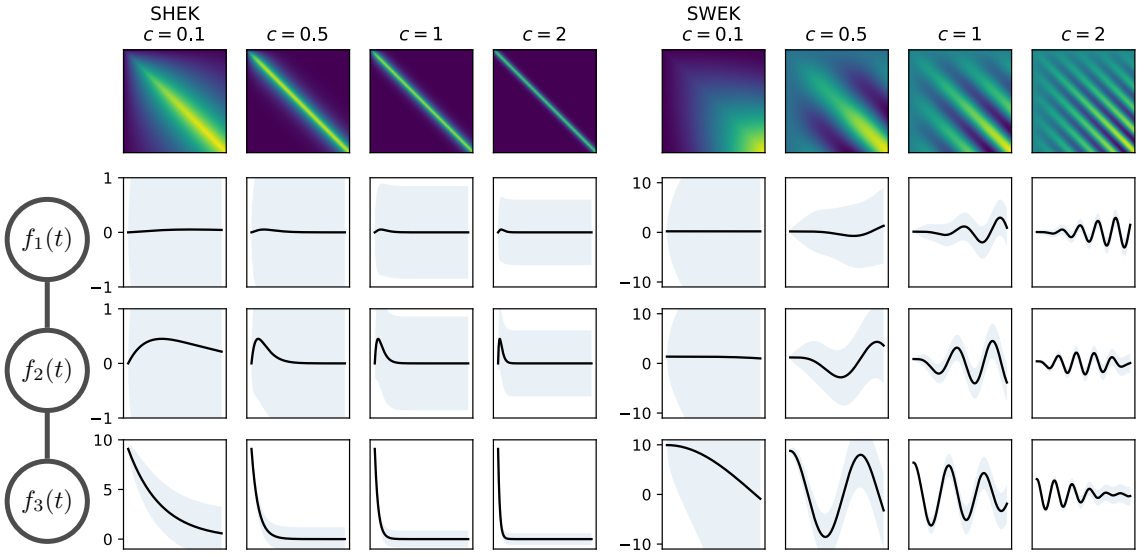
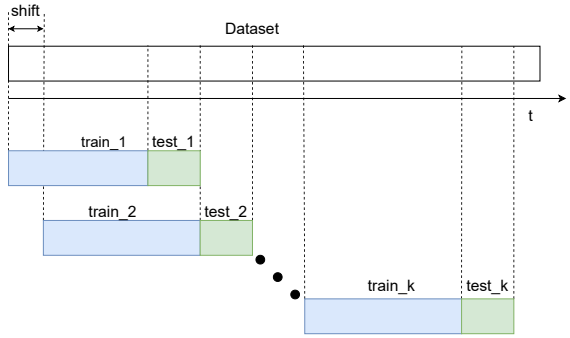


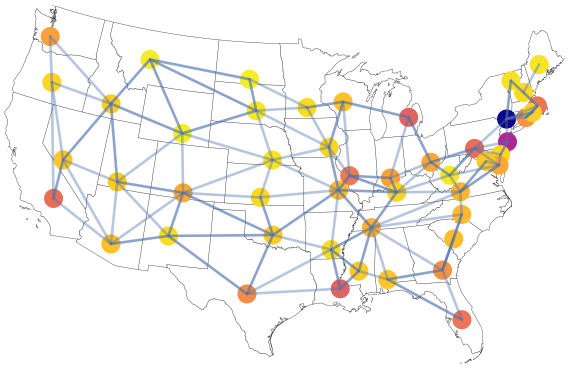
Figure S2: As in Fig. S1, here we illustrate mean and marginal variance.

E Experiment Details

We ran the experiments on a NVIDIA Tesla P100-PCIE-16GB GPU. We repeated the evaluations for several validation rounds (from eight to 12 depending on the experiment) using sliding window backtesting. Sliding window backtesting is schematically visualized in Figure S3a and explained in the caption. For all measurements, we report 95% confidence intervals.



(a) Sliding window backtesting visualization. At each step, we select a training time interval (▭) followed by a testing time interval (▫) and evaluate the performance. During the next iteration, we select time intervals that are shifted by a particular value.



(b) Visualization of the graph of adjacent states. Each node is a state in the US, and each edge indicates the adjacency between two states.

Figure S3

E.1 Synthetic Wave Experiments

We generated a wave distribution process over a one-dimensional line and discretized it with a graph with 11 vertices. We performed 12 iterations of sliding window backtesting on interpolation and extrapolation tasks. For the interpolation task, we used ten percent of randomly selected measurements over 52 timepoints. For the extrapolation task, we used 50 timepoints for training followed by two timepoints where we measured generalization error. We compared SHEK(ν=5/2) and SWEK(ν=5/2). We make conclusion that MAE is better for SWEK in interpolation (DM-test, $p < 0.01$) and extrapolation (DM-test, $p < 0.01$) tasks. The results are presented in Table S1.

Table S1: Interpolation/extrapolation performance for the wave distribution data over a line (MAE $\times$ 100).

Kernel	MAE_{int}	MAE_{ext}
SHEK(ν=5/2)	0.47 $\pm$ 0.14	2.74 $\pm$ 0.67
SWEK(ν=5/2)	**0.28 $\pm$ 0.06**	**1.89 $\pm$ 0.28**

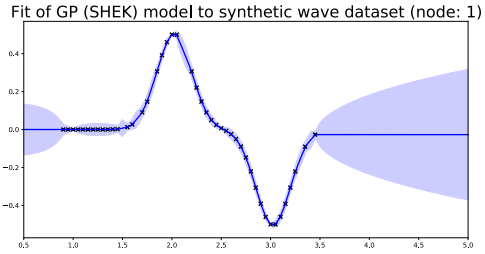
The difference between SHEK and SWEK can be seen visually in Figure S4. SWEK allows extrapolating wave behavior beyond training data, and SHEK, in contrast, generalizes badly on the synthetic wave dataset.

E.2 Chickenpox Experiments

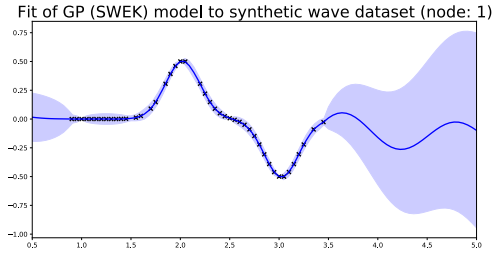
We performed sliding window backtesting for 12 iterations.

We report point-wise graphs that illustrate the comparison of generalization error on the extrapolation tasks with extended extrapolation periods in Figure S5. The statistical significance can be measured with the Diebold-Mariano test. For example, on four weeks extrapolation period, MAE of SHEK($\nu = 1/2$) is less than MAE of the product of Matérn(3/2) and RBF (DM-test, $p < 0.1$), as well as on six week extrapolation period (DM-test, $p < 0.01$).

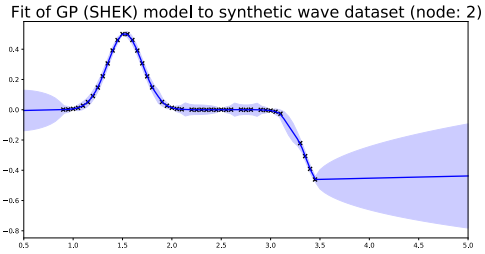
We also experimented with a larger number of separable kernels and compared them with each other and SHEK. For example, we provide results of four-weeks extrapolation in Figure S6.



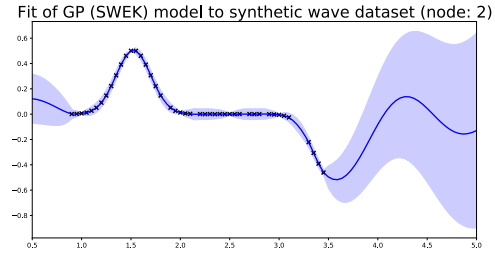
(a) SHEK on wave dataset (node #1).



(b) SWEK on wave dataset (node #1).

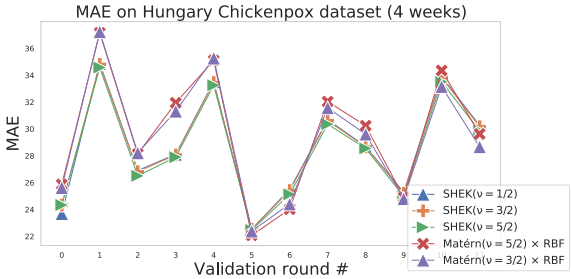


(c) SHEK on wave dataset (node #2).

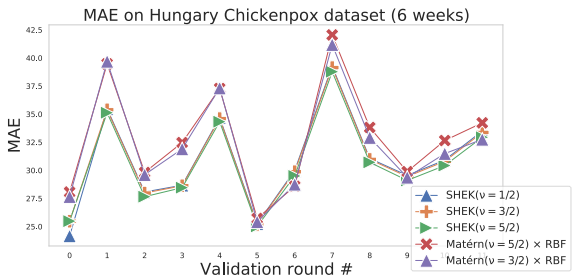


(d) SWEK on wave dataset (node #2).

Figure S4

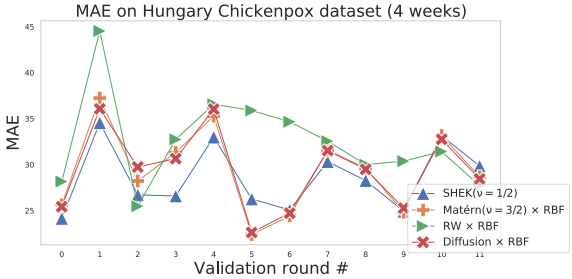


(a) Four weeks extrapolation.

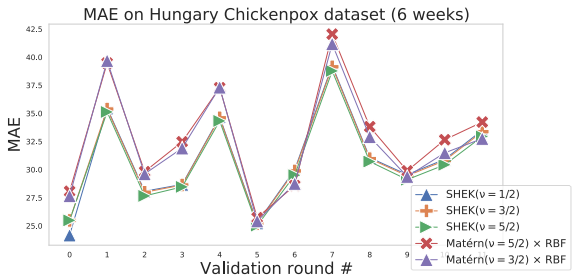


(b) Six weeks extrapolation.

Figure S5: Visualization of evaluation with extended extrapolation periods of four and six weeks on the Hungarian chickenpox dataset.



(a) Four weeks extrapolation (more separable kernels).

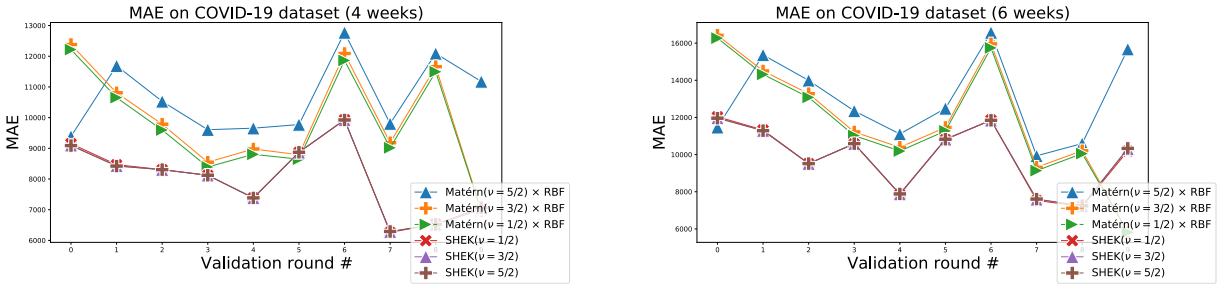


(b) Six weeks extrapolation (more separable kernels).

Figure S6: Visualization of evaluation with extended extrapolation periods of four and six weeks on the Hungarian chickenpox dataset (more separable kernels).

E.3 COVID-19 Experiments

The dataset for this use-case consisted of two parts: information about COVID-19 cases and deaths published by The New York Times The New York Times (2020), and a graph that was generated as follows. Each vertex represents a state, and two nodes v_1 and v_2 are connected if two states share a common border. The visualization of the graph is presented in Figure S3b.



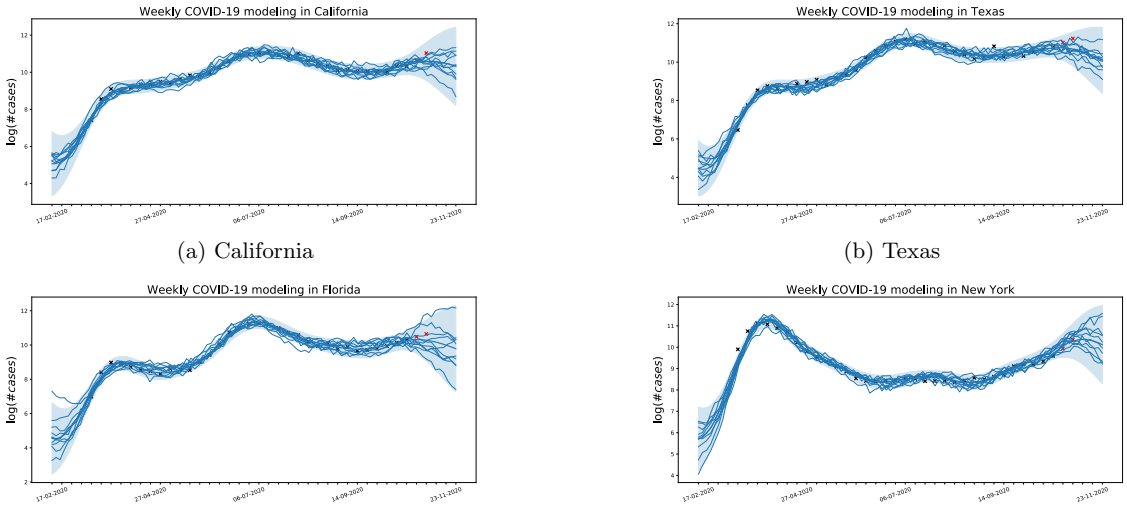
(a) Four weeks extrapolation.

(b) Six weeks extrapolation.

Figure S7: Visualization of extrapolation evaluation with extended extrapolation periods of four and six weeks on COVID-19 dataset.

We evaluated the performance of the model using ten runs with a sliding window backtesting. We used 33 weeks as training data and estimated the number of cases for the following two weeks. As a metric, we used mean absolute error (MAE).

We additionally report the results of extrapolation over four and six weeks periods of time in Figure S7. We can observe that the MAE of SHEK modifications is lower than separable Matérn kernels. It can be quantitatively measured using the Diebold-Mariano test. For example, four weeks period MAE of SHEK($\nu = 1/2$) is less than MAE of the product of Matérn(1/2) and RBF (DM-test, $p < 0.01$), as well as on six week period (DM-test, $p < 0.05$). Next, we provide the visualizations of fitted GP using the proposed graph kernel for the three states with the largest population in the USA. In Figure S8, we showed 97.5% confidence intervals provided by the GP that was trained on COVID-19 dataset.



(a) California

(b) Texas

(c) Florida

(d) New York

Figure S8: The visualizations of the GP posterior on the COVID-19 dataset for the most populated states. We indicate 97.5% confidence intervals with blue color; we also show samples from the posterior, and training and test data points (marked with black and red crosses, respectively).