Inferring Time Varying Signals over Uncertain Graphs

Sabbaqi, Mohammad; Isufi, Elvin

DOI
10.1109/ICASSP48485.2024.10446779

Publication date
2024

Document Version
Final published version

Published in
Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2024

ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-4485-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICASSP48485.2024.10446779

INFERRING TIME VARYING SIGNALS OVER UNCERTAIN GRAPHS

Mohammad Sabbaqi and Elvin Isufi

Delft University of Technology, Delft, The Netherlands

ABSTRACT

Inference of time varying data over graphs is of importance in real-world applications such as urban water networks, economics, and brain recordings. It typically relies on identifying a computationally affordable joint spatiotemporal method that can leverage the patterns in the data. While this per se is a challenging task, it becomes even more so when the network comes with uncertainties, which, if not accounted for, can lead to unpredictable consequences. To target this setting, we model graph uncertainties as Gaussian noise on the edges and design a stochastic partial differential equation (SPDE) based on it. We use this SPDE as a state equation to model the time varying signal evolution and extend it further to a state-space model where the observations are graph-filtered versions of the state. This allows us to have a joint spatiotemporal expressive kernel that can be estimated online via Kalman filtering and which parameters can also be estimated online via maximum likelihood principles, ultimately, reducing the computational cost. We corroborate the proposed approach on numerical experiments, showing a superior performance to approaches ignoring either the uncertainty or considering a separable spatiotemporal kernel.

Index Terms— Time-varying graph signals, Stochastic partial differential equations, Gaussian processes on graphs.

1. INTRODUCTION

Inference of time varying signals over graphs plays a key role in network-based systems to interpolate missing values, forecast a certain horizon, and detect anomalies, to name a few [1]. This is a challenging problem because the spatiotemporal coupling in the data needs to be exploited in a computationally affordable manner due to the problem dimensions (large graphs, long time horizon). This task becomes even more challenging when the graph is imperfect, which is typically the case in almost all physical networks (water, power, transport) but also when it is estimated from a finite amount of data [2, 3]. Topological uncertainties have been studied in different settings including graph filtering [4], graph signal processing [5], graph neural networks [6–9], and PDE-based approaches [10, 11].

THIS WORK IS SUPPORTED BY THE TU DELFT AI LABS PROGRAMME. E-MAILS: {M.SABBAQI; E.ISUFI-1}@TUDELFT.NL

A natural way to infer time varying signals on graphs is to extend graph kernels [12] into a spatiotemporal form, where due to computational aspects, a separable spatiotemporal kernel is favored [13, 14]. Another approach is to design a latent space model where independent Gaussian processes pass through a graph filter to account for both the temporal and the spatial connections in the data [15, 16]. The work in [17] also builds around Gaussian processes but designs a joint (non-separable) kernel based on SPDEs. The latter is more interpretable and expressive than the separable kernels but it is computationally heavier. In this work, a general diffusion matrix is considered to propagate the stochasticity which limits applicability to small graphs and short temporal windows.

To overcome the above, we propose a graph SPDE for modeling edge uncertainties leading to a joint spatiotemporal kernel for time varying data over the graph. The graph as an inductive bias structure rules the noise diffusion acting. We use this SPDE-based kernel in a state-space model to target more general graph processes that are not limited to low-pass behaviors on the graph. We approach the proposed model with Kalman filtering and maximum likelihood estimation to predict the process and estimate model parameters, ultimately, compensating for the graph uncertainties in a scalable setting. The proposed approach is tested numerically on three datasets to emphasize the importance of compensation for topological imperfections.

2. PROBLEM FORMULATION

Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ be an undirected graph with node set $\mathcal{V}$ of N nodes and edge set $\mathcal{E}$ of M edges. Let also $\mathbf{A}$ be the weighted adjacency matrix where entry $\mathbf{A}_{ij} = a_{ij} \geq 0$ indicates the weight of edge e_{ij}. We can represent the graph structure also via its incidence matrix $\mathbf{B} \in \mathbb{R}^{N\times M}$ that captures weighted proximities between nodes and edges. Consequently, the graph Laplacian is $\mathbf{L} = \text{diag}(\mathbf{A1}) - \mathbf{A} = \mathbf{B}\mathbf{B}^\top$, with $\mathbf{1}$ the all one vector and $\text{diag}(\cdot)$ the diagonalization operator. For completeness, we also denote the eigendecomposition of $\mathbf{L} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$, with eigenvectors across the columns of $\mathbf{V}$ and eigenvalues on the main diagonal of $\mathbf{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_N)$. One the nodes of $\mathcal{G}$, we consider a time-varying graph signal, which is a continuous mapping $f : \mathcal{V} \times \mathbb{R} \to \mathbb{R}$ that assigns a time series to each node. For discrete time instants, we can represent the signal in the matrix form $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_T] \in \mathbb{R}^{N\times T}$ where each snapshot

$\mathbf{x}_t \in \mathbb{R}^N$ is a graph signal.

2.1. Stochastic Partial Differential Equations

Since the Laplacian matrix discretizes the Laplace operator, a class of diffusion processes over graphs can be represented as a partial differential equation in terms of $\mathbf{L}$ [10, 18]. For instance, the popular heat diffusion kernel is a solution to

$$\mathrm{d}\mathbf{x} = -c\mathbf{L}\mathbf{x}\mathrm{d}t \tag{1}$$

where constant c describes the spatial diffusivity [19]. Building on this analogy, the work in [17] defined a heat SPDE to design joint spatiotemporal kernels for time varying data over networks. This SPDE reads as

$$\mathrm{d}\mathbf{x}_t = -c\mathbf{L}\mathbf{x}_t\mathrm{d}t + \mathbf{S}\mathrm{d}\boldsymbol{\beta}_t, \tag{2}$$

where $\boldsymbol{\beta}_t$ is a V-dimensional standard Brownian motion and $\mathbf{S} \in \mathbb{R}^{N\times V}$ is the diffusion matrix. Under Gaussian initial conditions $\mathbf{x}_0 \sim \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$, the solution of (2) is a Gaussian process of the form $\mathbf{x}_t \sim \mathcal{GP}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_{t,s}(\mathbf{L}))$ [17]. The mean process $\boldsymbol{\mu}_t$ is a heat diffusion on graphs $\boldsymbol{\mu}_t = \exp(-c\mathbf{L}t)\boldsymbol{\mu}_0$ with $\exp(\cdot)$ the matrix exponential. The covariance function is $\boldsymbol{\Sigma}_{t,s}(\mathbf{L}) = \mathbf{V}\mathbf{C}_{t,s}\mathbf{V}^\top$, where matrix $\mathbf{C}_{t,s}$ has (i,j)th entry

$$[\mathbf{C}_{t,s}]_{ij} = \frac{[\mathbf{V}^\top\mathbf{S}\mathbf{S}^\top\mathbf{V}]_{ij}}{c(\lambda_i+\lambda_j)}(e^{-c\lambda_i|t-s|} - e^{-c(\lambda_i t+\lambda_j s)}). \tag{3}$$

This covariance matrix defines a spatiotemporal kernel that suppresses higher graph frequencies and imposes low-pass properties. If the matrix $\mathbf{S}\mathbf{S}^\top$ is analytic in $\mathbf{L}$, this kernel shrinks into a spectral Gaussian process over the graph.

While the kernel in (3) has shown a great performance in interpolating and extrapolating networked time series, it has three main limitations. First, its complexity is of order cubic $\mathcal{O}(N^3T^3)$, limiting its applicability to small graphs. Second, its solution is limited to low-pass processes over graphs because of the first-order diffusion with $\mathbf{L}$. Third, the Brownian motion diffuses over the network regardless of the topology via a general diffusion matrix $\mathbf{S}$. Our goal is to propose an alternative to the SPDE (2) that overcomes all these issues.

2.2. Problem Motivation

We consider the observed graph $\mathbf{L}$ matches the support of the true graph $\mathbf{L}^\star$ but its edge weights differ slightly because of estimation error. This could be seen as a relative estimation error [4, 6, 20], and it is common in many applications involving physical networks such as water, transportation, and power networks. We model this error as

$$\mathbf{L}^\star = \mathbf{L} + \mathbf{B}\mathrm{diag}(\mathbf{w})\mathbf{B}^\top, \tag{4}$$

where $\mathbf{w} \in \mathbb{R}^M$ is a normal noise modeling the uncertainties over each edge independently. A heat diffusion model over the underlying graph can then be written as

$$\mathrm{d}\mathbf{x}_t = -c\mathbf{L}^\star\mathbf{x}_t\mathrm{d}t = -c\mathbf{L}\mathbf{x}_t\mathrm{d}t - c\mathbf{B}\mathrm{diag}(\mathbf{w})\mathbf{B}^\top\mathbf{x}_t\mathrm{d}t, \tag{5}$$

where process $\mathbf{x}_t$ is the superposition of the diffusion over the observed graph $\mathbf{L}$ and of the relative error graph $\mathbf{B}\mathrm{diag}(\mathbf{w})\mathbf{B}^\top$.

This SPDE is time-variant because of the term $\boldsymbol{\alpha}_t = \mathbf{B}^\top\mathbf{x}_t$. For processes $\mathbf{x}_t$ that are smooth over the graph in each t [21], we can consider $\boldsymbol{\alpha} = \boldsymbol{\alpha}_t$ and get the linear time-invariant (LTI)-SPDE

$$\mathrm{d}\mathbf{x}_t = -c\mathbf{L}\mathbf{x}_t\mathrm{d}t + \mathbf{B}\mathrm{diag}(\boldsymbol{\alpha})\mathrm{d}\boldsymbol{\beta}_t, \tag{6}$$

with $\mathrm{d}\boldsymbol{\beta}_t = \mathbf{w}\mathrm{d}t$. This means that the Brownian motion $\boldsymbol{\beta}_t$ has independent entries with different energies $\boldsymbol{\alpha}$ that depend on the graph signal differences $\mathbf{B}^\top\mathbf{x}_t$. Thus, the uncertainty will play a bigger role on edges with a bigger signal difference. Contrasting (6) with (2), we can also see that the noise diffusion matrix $\mathbf{S} = \mathbf{B}\mathrm{diag}(\boldsymbol{\alpha})$ is graph dependent, ultimately, allowing for a more interpretable model and less parameters to estimate –M values of $\boldsymbol{\alpha}$ in (6) instead of NV values of $\mathbf{S}$ in (2).

Model (6) is however a heat diffusion process and thus it is limited to low-pass graph signals at each snapshot. To achieve a more general version, we extend it to a state-space model where the state follows the LTI-SPDE (6) and the observation model follows the discrete-time graph filtering model

$$\mathbf{y}_k = \mathbf{M}(\tilde{\mathbf{H}}(\mathbf{L})\mathbf{x}_{t_k} + \tilde{\mathbf{v}}_k) := \mathbf{M}(\textstyle\sum_{i=0}^K h_i\mathbf{L}^i\mathbf{x}_{t_k} + \tilde{\mathbf{v}}_k), \tag{7}$$

where $\mathbf{y}_k$ is the observation at snapshot k, $\mathbf{x}_{t_k}$ is the sampled value of the continuous state variable $\mathbf{x}_t$, $\mathbf{M} \in \mathbb{R}^{F\times N}$ is a sampling matrix, and $\tilde{\mathbf{v}}_k \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_F)$ is the observation noise. Here, $\tilde{\mathbf{H}}(\mathbf{L}) := \sum_{i=0}^K h_i\mathbf{L}^i$ is the graph convolutional filtering matrix and expresses observation as a linear combination of the discretized states $\mathbf{x}_{t_k}$ from neighbors up to K hops away [21, 22]. We define $\mathbf{H} = \mathbf{M}\tilde{\mathbf{H}}$ for notational convenience.

The filter in (7) has $\mathcal{O}(K)$ parameters $\mathbf{h} = [h_0, \dots, h_K]^\top$ that are independent of the graph size and a computational complexity of order $\mathcal{O}(KM)$ [22]. These are favorable figures as they allow modeling a wide class of measurement models with affordable computations. Thus, with this state-space formulation, we can model time-varying data over graphs with arbitrary spectral behaviors in a recursive manner and a lower complexity as we shall detail next.

2.3. Problem Formulation

Consider the continuous-discrete state-space model

$$\begin{cases} \mathrm{d}\mathbf{x}_t = -c\mathbf{L}\mathbf{x}_t\mathrm{d}t + \mathbf{B}\mathrm{diag}(\boldsymbol{\alpha})\mathrm{d}\boldsymbol{\beta}_t \\ \mathbf{y}_k = \mathbf{H}(\mathbf{L})\mathbf{x}_{t_k} + \mathbf{v}_k \end{cases}, \tag{8}$$

with initial condition $\mathbf{x}_0 \sim \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$. Given a set of temporal observations $\mathbf{y}_1, \dots, \mathbf{y}_T$, our goal is to estimate the edge uncertainties $\boldsymbol{\alpha}$ and the filter coefficients $\mathbf{h}$ in a Gaussian process framework. This can be achieved by maximizing the likelihood function obtained from the solution of the state-space model. Given these parameters, $\mathbf{y}_k$ can be inferred via Kalman filtering.

Remark 1 (Gaussian process regression) *Since both the initial condition $\mathbf{x}_0$ and the Brownian motion $\boldsymbol{\beta}_t$ are Gaussian, the solution to the state equation* (6) *is a Gaussian process. Hence, the state-space model* (8) *can be indicated as $\mathbf{y}_k = \mathbf{H}(\mathbf{L})\mathbf{f}(\mathbf{x}_{t_k}) + \mathbf{v}_k$, where $\mathbf{f}(\cdot)$ is a non-separable spatiotemporal Gaussian process. In turn, this enables a Bayesian framework for parameter estimation.*

Remark 2 (Edge imperfections) *We compensate for the edge imperfections only in the state model (via estimating $\boldsymbol{\alpha}$) since it may lead to poor modeling performance due to temporally overgrowing error. Instead, we do not compensate for the edge imperfections in the observation equation as graph filters have shown to be stable to topological perturbations; see e.g., [4, 20, 23].*

3. INFERENCE ALGORITHM

Our inference algorithm consists of first discretizing the SPDE and then using Kalman filtering with an online parameter estimation via maximum likelihood.

3.1. Discretization of Continuous State

Given data with a high temporal resolution and uniform intervals $\triangle t$, the continuous state can be discretized with a negligible error. Since the transition matrix of the LTI-SPDE (6) is an exponential function $\exp(-c\mathbf{L}(t-s))$, we can discretize (6) by multiplying both sides with the transition matrix and integrating over interval $[0, \triangle t]$ [24]. The discretization yields the state equation

$$\mathbf{x}_{t_k+\triangle t} = \tilde{\mathbf{L}}\mathbf{x}_{t_k} + \mathbf{q}_k, \qquad \mathbf{q}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}) \tag{9}$$

with transition operators

$$\tilde{\mathbf{L}} = \exp(-c\Delta t\mathbf{L}) \tag{10a}$$

$$\mathbf{Q} = \int_0^{\Delta t} e^{-c\mathbf{L}(\Delta t-s)}\mathbf{B}\mathrm{diag}^2(\boldsymbol{\alpha})\mathbf{B}^\top e^{-c\mathbf{L}(\Delta t-s)}\, ds \tag{10b}$$

where $\mathrm{diag}^2(\cdot)$ stands in for the square of the diagonal matrix [25]. Using the *matrix fraction* method in [26], we can solve the integral in (10b) by first defining the matrix

$$\exp\left(\begin{bmatrix} c\mathbf{L} & \mathbf{B}\mathrm{diag}(\boldsymbol{\alpha}^2)\mathbf{B}^\top \\ \mathbf{0} & -c\mathbf{L}\end{bmatrix}\triangle t\right) = \begin{bmatrix}\mathbf{F}_1 & \mathbf{G} \\ 0 & \mathbf{F}_2\end{bmatrix}, \tag{11}$$

and then setting the solution to $\mathbf{Q} = \mathbf{F}_2^\top\mathbf{G}$.

There are multiple approaches to compute a matrix exponential [27] but since the data is available with a relatively high resolution (small $\triangle t$), we can rely on a first-order Taylor approximation for both equations (10a) and (11). Hence, we get the approximations [1]

$$\tilde{\mathbf{L}} \simeq \mathbf{I} - c\triangle t\mathbf{L} \tag{12a}$$

$$\mathbf{Q} \simeq \triangle t\mathbf{B}\mathrm{diag}^2(\boldsymbol{\alpha})\mathbf{B}^\top. \tag{12b}$$

[1] We have also tested the second-order Taylor approximation for these matrix exponentials but they lead to numerical instabilities due to $\mathbf{H}^{-1}(\mathbf{L})$.

Equation 12b suggests that the covariance matrix of the system noise has rank M. It is parameterized by $\boldsymbol{\alpha}$, and accounts for the graph structure. Bringing all together, we obtain the discrete-time state-space model

$$\mathbf{x}_k = \tilde{\mathbf{L}}\mathbf{x}_{k-1} + \mathbf{q}_k, \quad \mathbf{y}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k, \tag{13}$$

where the covariance matrix of the system and observation noises are $\mathbf{Q} = \mathbf{B}\mathrm{diag}^2(\boldsymbol{\alpha})\mathbf{B}^\top$ and $\mathbf{R} = \sigma^2\mathbf{I}$, respectively. Note that we merged $\triangle t$ with $\boldsymbol{\alpha}$ in $\mathbf{Q}$ for notation convenience.

3.2. Parameter Estimation

Given the model parameters, $\boldsymbol{\alpha}$, σ^2, and $\mathbf{H}(\mathbf{L})$, the solution of state space model (13) can be obtained via discrete Kalman filtering. Given the updates up to iteration $k-1$ of the state variable $\mathbf{x}_{k-1}^{k-1} = \mathbb{E}[\mathbf{x}_{k-1}|\mathbf{y}_1, \ldots, \mathbf{y}_{k-1}]$ and the respective covariance matrix $\mathbf{P}_{k-1}^{k-1} = \mathrm{Cov}[\mathbf{x}_{k-1}|\mathbf{y}_1, \ldots, \mathbf{y}_{k-1}]$ the Kalman updates can be computed as:
Prediction step:

$$\mathbf{x}_k^{k-1} = \tilde{\mathbf{L}}\mathbf{x}_{k-1}^{k-1}; \tag{14a}$$

$$\mathbf{P}_k^{k-1} = \tilde{\mathbf{L}}\mathbf{P}_{k-1}^{k-1}\tilde{\mathbf{L}} + \mathbf{B}\mathrm{diag}^2(\boldsymbol{\alpha})\mathbf{B}^\top; \tag{14b}$$

Correction step:

$$\mathbf{K}_k = \mathbf{P}_k^{k-1}\mathbf{H}^\top(\mathbf{H}\mathbf{P}_k^{k-1}\mathbf{H}^\top + \sigma^2\mathbf{I})^{-1} \tag{15a}$$

$$\mathbf{x}_k^k = \mathbf{x}_k^{k-1} + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}\mathbf{x}_k^{k-1}); \tag{15b}$$

$$\mathbf{P}_k^k = \mathbf{P}_k^{k-1} - \mathbf{K}_k\mathbf{P}_k^{k-1}\mathbf{K}_k^\top; \tag{15c}$$

where $\mathbf{K}_k$ is the Kalman gain at iteration k. Due to its online nature, the Kalman filter has an order of complexity that is linear in time $\mathcal{O}(N^3T)$.

Running the Kalman filter (14a)-(15c) with fixed parameters leads to the distribution

$$p(\mathbf{y}_k|\mathbf{y}_1, \ldots, \mathbf{y}_{k-1}) = \mathcal{N}(\mathbf{H}\mathbf{x}_k^{k-1}, \mathbf{H}\mathbf{P}_k^{k-1}\mathbf{H}^\top + \sigma^2\mathbf{I}) \tag{16}$$

which allows us to compute the marginal likelihood. The negative log-likelihood function can be computed recursively as

$$\begin{aligned}\mathcal{L}_k(\boldsymbol{\alpha}, \mathbf{h}) = {}& \mathcal{L}_{k-1}(\boldsymbol{\alpha}, \mathbf{h}) + \tfrac{1}{2}\log|\mathbf{S}_k| \\ & + \tfrac{1}{2}(\mathbf{y}_k - \mathbf{H}\mathbf{x}_k^{k-1})^\top\mathbf{S}_k^{-1}(\mathbf{y}_k - \mathbf{H}\mathbf{x}_k^{k-1})\end{aligned} \tag{17}$$

with $\mathbf{S}_k = \mathbf{H}\mathbf{P}_k^{k-1}\mathbf{H}^\top + \sigma^2\mathbf{I}$ and $|\cdot|$ denoting the matrix determinant. We can then obtain the estimates for $\boldsymbol{\alpha}$ and $\mathbf{h}$ via a simple alternating minimization on $\mathcal{L}_k(\boldsymbol{\alpha}, \mathbf{h})$ first on $\boldsymbol{\alpha}$ and then on $\mathbf{h}$. For this, we use online gradient descent, where a gradient update occurs at each time stamp. As the loss function is written recursively, all the updates can also be computed recursively as

$$\begin{aligned}\boldsymbol{\alpha}_k &= \boldsymbol{\alpha}_{k-1} - \gamma_1\nabla_{\boldsymbol{\alpha}}\mathcal{L}_k(\boldsymbol{\alpha}_{k-1}, \mathbf{h}_{k-1}), \\ \mathbf{h}_k &= \mathbf{h}_{k-1} - \gamma_2\nabla_{\mathbf{h}}\mathcal{L}_k(\boldsymbol{\alpha}_k, \mathbf{h}_{k-1}).\end{aligned} \tag{18}$$

Table 1. Interpolation task performance for both synthetic and weather temperature dataset. The experiments are performed based on different portions of unobserved data.

rNMSE	Synthetic			Weather			Traffic		
	10%	20%	30%	10%	20%	30%	10%	20%	30%
LMS	0.40	0.46	0.46	0.42	0.43	0.49	0.41	0.45	0.48
StarGP	0.31	0.31	0.36	0.25	0.24	0.31	0.21	0.25	0.29
G-SPDE	**0.12**	**0.14**	**0.16**	**0.13**	**0.14**	**0.17**	**0.16**	**0.15**	**0.27**
No-α	0.24	0.27	0.30	0.23	0.28	0.31	0.27	0.33	0.37
Fixed-α	0.24	0.26	0.29	0.22	0.26	0.29	0.24	0.27	0.36
Learn-S	0.23	0.20	0.20	0.21	0.21	0.27	0.19	0.22	0.34

The gradient can be computed in closed form but its explicit expression is omitted here for the sake of space.

Altogether, the proposed approach consists of modeling the spatiotemporal process via the state-space model (8) where the state equation is an LTI-SPDE that accounts for the edge uncertainties and the observations are discrete-time signals that are aggregated by a graph convolutional filter. Upon discretizing the state-space model (13), we solve it recursively by alternating between the Kalman filtering updates in (14a)-(15c) to infer the spatiotemporal data and the maximum likely estimates of the parameters in (18) to compensate for the uncertainties. We conclude with the following remark.

Remark 3 (Maximum a posteriori) *If prior information over $\boldsymbol{\alpha}$ or $\mathbf{h}$ is available, we can add to* (17) *the negative log-priors $-\log p_\alpha(\boldsymbol{\alpha})$ and $-\log p_h(\mathbf{h})$. We can consider the Gaussian prior $\boldsymbol{\alpha} \sim \mathcal{N}(\mathbf{0}, \sigma_p^2 \text{diag}(\tilde{\mathbf{B}}^\top \mathbf{d}))$ where $\tilde{\mathbf{B}}$ is the unweighted and undirected incidence matrix and $\mathbf{d}$ is the degree vector. This means that edges connected to nodes with higher degrees are more susceptible to uncertainty which is closely tied to relative perturbations over graphs.*

4. NUMERICAL EXPERIMENTS

We corroborate the proposed graph SPDE (G-SPDE) approach on spatiotemporal interpolation and extrapolation (forecasting) tasks on the following three datasets:

- **Synthetic**: A numerically generated SPDE over a stochastic block model of 200 nodes and 4 communities with inter/intra community probabilities of 0.8/0.2, and 10000 snapshots. The observation noise energy is $\sigma^2 = 0.01$. Both $\boldsymbol{\alpha}$ and $\mathbf{h}$ are multivariate normal and we report the average performance over different realizations.
- **NOAA**: Contains 8579 hourly temperatures across 109 stations in the U.S. and we used the setting in [28].
- **METR-LA**: A traffic dataset of four months with a 5-minute resolution over 207 nodes in Los Angeles. The graph is an exponential kernel of distance matrix as in [6].

In all the experiments, we split the datasets temporally into 10 parts and reported the average performance of the models. In each experiment, up to the starting 10% of the data are used for training and stabilizing parameter estimation. The constants $c = 0.75$ and $\triangle t = 1$ are considered throughout the experiments. For the interpolation task, we consider the prediction (14a) as the reconstructed state and pass it through the graph filter to obtain the missing values. For the extrapolation task, we forecast the trajectory via Kalman equations with the estimated parameters $\boldsymbol{\alpha}$ and $\mathbf{h}$ using a random noise generator with variance σ^2 in equation (15b). All the initial mean values and the initial maximum likelihood function in (17) are zero, while the initial covariance matrix $\boldsymbol{\Sigma}_0$ comes from Riccati's equation as the steady-state solution of ordinary differential equation [25]. As alternatives, we consider the standard least mean square (LMS) approach and StarGP from [15] that is based on Gaussian processes. The hyperparameters for these baselines are selected based on [15].

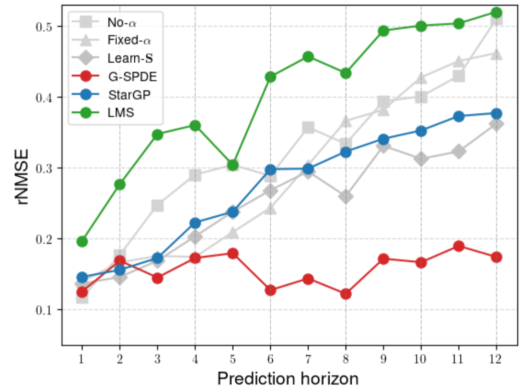

Fig. 1. Traffic forecasting performance in rNMSE for proposed models with different prediction horizons.

The top three lines of Table 1 and Fig. 1 compare the methods for the interpolation and the extrapolation task, respectively. We see that the proposed approach outperforms the alternatives by a margin and this is attributed to the fact that we compensate for the graph perturbations. We further investigate the role of the different components in our method: i) No-$\boldsymbol{\alpha}$ is the model ignoring edge uncertainties; ii) Fixed-$\boldsymbol{\alpha}$ assumes the noise energy is known; and Learn-$\mathbf{S}$ ignores graph structure and estimates a matrix $\mathbf{S}$. In all cases, we see that imposing a graph structure in $\mathbf{S}$ and compensating for the uncertainties reduces drastically the estimation error.

5. CONCLUSION

This paper proposed a method to infer spatiotemporal graph signals on networks with edge uncertainties. We consider a state-space model, where the state equation is based on stochastic partial differential equations tailored to the graph perturbations, and the observation matrix is a graph filter. By leveraging a Kalman filter approach we solved the state-space model and used its prediction distribution to estimate both the covariance matrix of the edge uncertainties and the filter coefficients in an online fashion. The latter allows for compensating the graph uncertainties. The proposed approach can also be seen as having a joint spatiotemporal kernel over the graph as a latent variable where the observations are obtained through a graph filter. Experimental results showcase the superiority of the proposed method compared with alternative solutions.

6. REFERENCES

[1] E. D. Kolaczyk, G. Csárdi, E. D. Kolaczyk, and G. Csárdi, “Statistical models for network graphs,” *Statistical analysis of network data with R*, 2014.

[2] A. Buciulea, S. Rey, and A. G. Marques, “Learning graphs from smooth and graph-stationary signals with hidden variables,” *IEEE Transactions on Signal and Information Processing over Networks*, 2022.

[3] Y. Shen, B. Baingana, and G. B. Giannakis, “Kernel-based structural equation models for topology identification of directed networks,” *IEEE Transactions on Signal Processing*, 2017.

[4] F. Gama, J. Bruna, and A. Ribeiro, “Stability properties of graph neural networks,” *IEEE Transactions on Signal Processing*, 2020.

[5] E. Ceci and S. Barbarossa, “Graph signal processing in the presence of topology uncertainties,” *IEEE Transactions on signal processing*, 2020.

[6] M. Sabbaqi and E. Isufi, “Graph-time convolutional neural networks: Architecture and theoretical analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[7] Z. Gao and E. Isufi, “Learning stochastic graph neural networks with constrained variance,” *IEEE Transactions on Signal Processing*, 2023.

[8] Z. Gao, E. Isufi, and A. Ribeiro, “Stochastic graph neural networks,” *IEEE Transactions on Signal Processing*, 2021.

[9] ——, “Stability of graph convolutional neural networks to stochastic perturbations,” *Signal Processing*, 2021.

[10] B. Chamberlain, J. Rowbottom, M. I. Gorinova, M. Bronstein, S. Webb, and E. Rossi, “Grand: Graph neural diffusion,” in *International Conference on Machine Learning*. PMLR, 2021.

[11] Y. Song, Q. Kang, S. Wang, K. Zhao, and W. P. Tay, “On the robustness of graph neural diffusion to topology perturbations,” *Advances in Neural Information Processing Systems*, 2022.

[12] A. J. Smola and R. Kondor, “Kernels and regularization on graphs,” in *Learning Theory and Kernel Machines: 16th Annual Conference on Learning Theory and 7th Kernel Workshop, COLT/Kernel 2003, Washington, DC, USA, August 24-27, 2003. Proceedings*. Springer, 2003.

[13] D. Romero, V. N. Ioannidis, and G. B. Giannakis, “Kernel-based reconstruction of space-time functions on dynamic graphs,” *IEEE Journal of Selected Topics in Signal Processing*, 2017.

[14] Q. Lu and G. B. Giannakis, “Probabilistic reconstruction of spatio-temporal processes over multi-relational graphs,” *IEEE Transactions on Signal and Information Processing over Networks*, 2021.

[15] ——, “Spatio-temporal inference of dynamical gaussian processes over graphs,” in *2021 55th Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2021.

[16] Y.-C. Zhi, Y. C. Ng, and X. Dong, “Gaussian processes on graphs via spectral kernel learning,” *IEEE Transactions on Signal and Information Processing over Networks*, 2023.

[17] A. V. Nikitin, S. John, A. Solin, and S. Kaski, “Non-separable spatio-temporal graph kernels via spdes,” in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2022.

[18] M. Wardetzky, “Discrete laplace operators,” *An Excursion Through Discrete Differential Geometry: AMS Short Course, Discrete Differential Geometry*, 2020.

[19] D. Thanou, X. Dong, D. Kressner, and P. Frossard, “Learning heat diffusion graphs,” *IEEE Transactions on Signal and Information Processing over Networks*, 2017.

[20] R. Levie, E. Isufi, and G. Kutyniok, “On the transferability of spectral graph filters,” in *2019 13th International conference on Sampling Theory and Applications (SampTA)*. IEEE, 2019.

[21] R. Ramakrishna, H.-T. Wai, and A. Scaglione, “A user guide to low-pass graph signal processing and its applications: Tools and applications,” *IEEE Signal Processing Magazine*, 2020.

[22] E. Isufi, F. Gama, D. I. Shuman, and S. Segarra, “Graph filters for signal processing and machine learning on graphs,” *arXiv preprint arXiv:2211.08854*, 2022.

[23] H. Kenlay, D. Thanou, and X. Dong, “Interpretable stability bounds for spectral graph filters,” in *International conference on machine learning*. PMLR, 2021.

[24] P. Axelsson and F. Gustafsson, “Discrete-time solutions to the continuous-time differential lyapunov equation with applications to kalman filtering,” *IEEE Transactions on Automatic Control*, 2014.

[25] S. Särkkä and A. Solin, *Applied stochastic differential equations*. Cambridge University Press, 2019.

[26] C. Van Loan, “Computing integrals involving the matrix exponential,” *IEEE transactions on automatic control*, 1978.

[27] C. Moler and C. Van Loan, “Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later,” *SIAM review*, 2003.

[28] E. Isufi, A. Loukas, N. Perraudin, and G. Leus, “Forecasting time series with varma recursions on graphs,” *IEEE Transactions on Signal Processing*, 2019.