Hamelijnck, Oliver; Wilkinson, William; Loppi, Niki; Solin, Arno; Damoulas, Theodoros

**Spatio-temporal variational Gaussian processes**

Published in:
Advances in Neural Information Processing Systems 34 pre-proceedings (NeurIPS 2021)

Published: 01/01/2021

**Document Version**
Publisher's PDF, also known as Version of record

**Please cite the original version:**
Spatio-Temporal Variational Gaussian Processes

Oliver Hamelijnck*  
The Alan Turing Institute / University of Warwick  
ohamelijnck@turing.ac.uk

William J. Wilkinson*  
Aalto University  
william.wilkinson@aalto.fi

Niki A. Loppi  
NVIDIA  
nloppi@nvidia.com

Arno Solin  
Aalto University  
arno.solin@aalto.fi

Theodoros Damoulas  
The Alan Turing Institute / University of Warwick  
tdamoulas@turing.ac.uk

Abstract

We introduce a scalable approach to Gaussian process inference that combines spatio-temporal filtering with natural gradient variational inference, resulting in a non-conjugate GP method for multivariate data that scales linearly with respect to time. Our natural gradient approach enables application of parallel filtering and smoothing, further reducing the temporal span complexity to be logarithmic in the number of time steps. We derive a sparse approximation that constructs a state-space model over a reduced set of spatial inducing points, and show that for separable Markov kernels the full and sparse cases exactly recover the standard variational GP, whilst exhibiting favourable computational properties. To further improve the spatial scaling we propose a mean-field assumption of independence between spatial locations which, when coupled with sparsity and parallelisation, leads to an efficient and accurate method for large spatio-temporal problems.

1 Introduction

Most real-world processes occur across space and time, exhibit complex dependencies, and are observed through noisy irregular samples. Take, for example, the task of modelling air pollution across a city. Such a task involves large amounts of noisy, partially-observed data with strong seasonal effects governed by weather, traffic, human movement, etc. This setting motivates a probabilistic perspective, allowing for the incorporation of prior knowledge and the quantification of uncertainty.

Gaussian processes (GPs, [38]) provide such a probabilistic modelling paradigm, but their inherent cubic computational scaling in the number of data, $N$, limits their applicability to spatio-temporal tasks. Arguably the most successful methods to address this issue are sparse GPs [37], which summarise the true GP posterior through a reduced set of $M$ inducing points and have dominant computational scaling $\mathcal{O}(NM^2)$, and spatio-temporal GPs [43], which rewrite the GP prior as a state-space model and use filtering to perform inference in $\mathcal{O}(Nd^3)$, where $d$ is the dimensionality of the state-space. Sparse GPs and spatio-temporal GPs have been combined by constructing a Markovian system in which a set of spatial inducing points are tracked over time [24, 51].

However, existing methods for spatio-temporal GPs make approximations to the prior conditional model [24] or do not exploit natural gradients [45], meaning they do not provide the same inference and learning results as state-of-the-art variational GPs [26] in the presence of non-conjugate likelihoods or sparsity, which has hindered their widespread adoption. We introduce spatio-temporal
variational GPs (ST-VGP), which provide the exact same results as standard variational GPs, whilst reducing the computational scaling in the temporal dimension from cubic to linear. ST-VGP is derived using a natural gradient variational inference approach based on filtering and smoothing. We also derive this method’s sparse variant, and demonstrate how it enables the use of significantly more inducing points than the standard approach, leading to improved predictive performance.

We then show how the spatio-temporal structure can be exploited even further to improve both the temporal and spatial scaling. We demonstrate for the first time how to apply parallel filtering and smoothing [41] to non-conjugate GPs to reduce the temporal (span) complexity to be logarithmic. We then reformulate the model to enable an efficient mean-field approximation across space, improving the complexity with respect to the number of spatial points. We analyse the practical performance and scalability of our proposed methods, demonstrating how they make it possible to apply GPs to large-scale spatio-temporal scenarios without sacrificing inference quality.

1.1 Related Work

GPs are commonly used for spatio-temporal modelling in both machine learning and spatial statistics [38, 22, 14, 6]. Many approaches to overcome their computational burden have been proposed, from nearest neighbours [17] to parallel algorithms on GPUs [48]. Within machine learning, the sparse GP approach is perhaps the most popular [37, 46], and is typically combined with mini-batching to allow training on massive datasets [26]. However, it fails in practical cases where the number of inducing points must grow with the size of the data, such as for time series [49].

When the data lie on a grid, separable kernels exhibit Kronecker structure which can be exploited for efficient inference [39]. This approach has been generalised to the partial grid setting [53], and to structured kernel interpolation (SKI, [52]) which requires only that inducing points be on a grid. Generally, these approaches are limited to the conjugate case, although Laplace-based extensions exist [19]. Bruinsma et al. [10] present an approach to spatio-temporal modelling that performs an orthogonal projection of the data to enforce independence between the latent processes.

It has been shown that variational GPs can be computed in linear time either by exploiting sparse precision structure [18] or via filtering and smoothing [11]. Other inference schemes such as Laplace and expectation propagation have also been proposed [35, 51]. In the spatio-temporal case, sparsity has been used in the spatial dimension [24, 43]. These methods historically suffered from the fact that i) filtering was not amenable to fast automatic differentiation due to its recursive nature, and ii) state-of-the-art inference schemes had not been developed to make them directly comparable to other methods. The first is no longer an issue since many machine learning frameworks are now capable of efficiently differentiating recursive models [11]. We address the second point with this paper. A similar algorithm to ours that is also sparse in the temporal dimension has been developed [50, 2], and relevant properties of the spatio-temporal model presented here are also analysed in [45]. Fourier features [28] are an alternative approach to scalable GPs, but are not suited to very long time series with high variability due to the need for impractically many inducing features.
2 Background

We consider data lying on a spatio-temporal grid comprising input–output pairs, \( \{ X^{(st)} \} \in \mathbb{R}^{N_t \times N_s \times D}, Y^{(st)} \in \mathbb{R}^{N_t \times N_s} \), where \( N_t \) is the number of temporal points, \( N_s \) the number of spatial points, and \( D = 1 + D_x \) the input dimensionality (with \( D_x \) being the number of spatial dimensions). We use \( t \) and \( s \) to represent time and space respectively. The assumption of the grid structure is relaxed via the introduction of sparse methods in Sec. 3, and by the natural handling of missing data.

For consistency with the GP literature we let \( X = \text{vec}(X^{(st)}) \in \mathbb{R}^{N \times D}, Y = \text{vec}(Y^{(st)}) \in \mathbb{R}^{N \times 1} \), where \( N = N_t N_s \) is the total number of data points. We use the operator \( \text{vec}(\cdot) \) to simply convert data from a spatio-temporal grid into vector form, whilst keeping observations ordered by time and then space. For notational convenience we define \( X_{n,k} = X^{(st)}_{n,k}, Y_{n,k} = Y^{(st)}_{n,k} \), which indexes data at time index \( n \) and spatial index \( k \). We use \( t_n \) to denote the \( n \)th time point, \( S \in \mathbb{R}^{N_t \times D_x} \) to denote all spatial grid points and \( S_k \) the \( k \)th one. Let \( f : \mathbb{R}^{D} \rightarrow \mathbb{R} \) to be a random function with a zero-mean GP prior, then for a given likelihood \( p(Y \mid f(X)) \) the generative model is,

\[
\int \cdots \int \mathcal{N}(0, \kappa(x, x')) \mathcal{N}(y, \beta) \exp(p(Y \mid f(X))) \, dx \, dx',
\]

where \( f_{n,k} = f(X_{n,k}) \), and we let \( f_n \) be the function values of all spatial points at time \( t_n \). When the kernel \( \kappa \) is evaluated at given inputs we write the corresponding gram matrix as \( K_{XX} = \kappa(X, X') \).

To make it explicit that \( f \) takes spatio-temporal inputs we also abuse the notation slightly to write \( f(x) = f(t, s) \) and \( \kappa(x, x') = \kappa(t, s, t', s') \). A summary of all notation used is provided in App. A. For Gaussian likelihoods the posterior, \( p(f \mid Y) \), is available in closed form, otherwise approximations must be used. In either case, inference typically comes at a cubic cost of \( \mathcal{O}(N_t^3 N_s^3) \).

2.1 State Space Spatio-Temporal Gaussian Processes

One method for handling the cubic scaling of GPs is to reformulate the prior in Eq. (1) as a state space model, reducing the computational scaling to linear in the number of time points [43]. The enabling assumption is that the kernel is both Markovian and separable between time and space: \( \kappa(t, s, t', s') = \kappa_t(t, t') \kappa_s(s, s') \). We use the term Markovian kernel to refer to a kernel which can be re-written in state-space form (see [44] for an overview). First, we write down the GP prior as a stochastic partial differential equation (SPDE, see [15]) \( \partial_t f(t, s) = A_n f(t, s) + L_n w(t, s) \), where \( w(t, s) \) is a (spatio-temporal) white noise process and \( A_n \) a suitable (pseudo-)differential operator [see 42]. By appropriately defining the model matrices and the white noise spectral density function, SPDEs of this form can represent a large class of separable and non-separable GP models.

When the kernel is separable, this SPDE can be simplified to a finite-dimensional SDE [24] by marginalising to a finite set of spatial locations, \( S \in \mathbb{R}^{N_s \times D_x} \) giving, \( \partial_t f(t) = F f(t) + L \delta \beta(t), \) where \( f(t) \) is the Gaussian distributed state at the spatial points \( S \) at time \( t \), with dimensionality \( d = N_s d_t \), where \( d_t \) is the dimensionality of the state-space model induced by \( \kappa_t(\cdot, \cdot) \). \( \delta \beta(t) \) has spectral density \( Q_n \), and the matrix \( H \) extracts the function value from the state: \( f_n = H f(t_n) \). and \( L \) are the feedback and noise effect matrices [42]. This simplification to an SDE is possible due to the independence between spatial points at time \( t \) and all other time steps, given the current state [45]. This follows from the fact that for any separable kernel, \( f(t, s) \) and \( f(t', s') \) are independent given \( f(t', s) \) [36]. For a step size \( \Delta_n = t_{n+1} - t_n \), the discrete-time model matrices are,

\[
A_n = \Phi(\Delta_n), \quad Q_n = \int_0^{\Delta_n} \Phi(\Delta_n - \tau) L Q_n L^\top \Phi(\Delta_n - \tau) \, d\tau,
\]

where \( \Phi(\cdot) \) is the matrix exponential. The resulting discrete model is,

\[
\tilde{f}(t_{n+1}) = A_n \tilde{f}(t_n) + q_n, \quad \hat{Y}_n \mid \tilde{f}(t_n) \sim p(Y_n \mid H f(t_n)),
\]

where \( q_n \sim \mathcal{N}(0, Q_n) \). If \( p(Y_n \mid H f(t_n)) \) is Gaussian then Kalman smoothing algorithms can be employed to perform inference in Eq. (3) in \( \mathcal{O}(N_t d^3) = \mathcal{O}(N_t N_s^3 d_t^3) \).

Markovian GPs with Spatial Sparsity - Sparse GPs re-define the GP prior over a smaller set of \( M \) inducing points: let \( u = f(Z) \in \mathbb{R}^{M \times D} \) be the inducing variables at inducing locations \( Z \in \mathbb{R}^{M \times D} \), then the augmented prior is \( p(f, u) = p(f \mid u) p(u) \), where \( p(u) = \mathcal{N}(u \mid 0, K_{ZZ}) \), and with Gaussian conditional \( p(f \mid u) \). If the inducing points are placed on a spatio-temporal grid, with \( Z_{ss} \in \mathbb{R}^{M_s \times D_s} \) being the spatial inducing locations, the conditional \( p(f \mid u) \) can be simplified to (see App. D):

\[
p(f \mid u) = \mathcal{N}(f \mid \left[ \mathbb{I} \otimes (K_{SS} + K_{Z_s Z_s}) \right] u, K_{tt} \otimes \hat{Q}_n),
\]
where $\bar{Q}_s = \mathbf{K}^{(s)}_{SS} - \mathbf{K}^{(s)}_{SZ} \mathbf{K}^{-1}_{zz} \mathbf{K}^{(s)}_{Z,S}$ (see App. A for notational details). The fully independent training conditional (FITC) method [37] approximates the full conditional covariance with its diagonal, leading to the following convenient property: $q_{\text{FITC}}(f \mid u) = \prod_{n=1}^{N_i} q_{\text{FITC}}(f_n \mid u_n)$, where the last equality holds because $I \otimes \left( \mathbf{K}^{(s)}_{SS} \otimes \mathbf{K}^{-1}_{zz} \right)$ is block diagonal. This factorisation across time allows the model to be cast into the state-space form of Eq. (3), but where the state $f(t)$ is defined over the reduced set of spatial inducing points [24]. Inference can be performed in $O(N_i M d_f^2)$.  

2.2 Sparse Variational GPs

To perform approximate inference in the presence of sparsity or non-Gaussian likelihoods, variational methods cast inference as optimisation through minimisation of the Kullback–Leibler divergence (KLD) from the true posterior to the approximate posterior [8]. Although direct computation of the KLD is intractable, it can be rewritten as the maximisation of the evidence lower bound (ELBO). To re-emphasise that the CVI updates compute the exact same quantity as Eq. (6), we provide a table of mappings between the various parametrisations is given in App. G.

Natural Gradients Natural gradient descent calculates gradients in distribution space rather than parameter space, and has been shown to improve inference time and quality for variational GPs [26, 40]. A natural descent direction is obtained by scaling the standard gradient by the inverse of the Fisher information matrix, $\mathbb{E}_{q(\cdot)} \left[ \nabla^2 \log q(\cdot) \right]$ [5]. For a Gaussian approximate posterior, the natural gradient of target $\mathcal{L}$ with respect to the natural parameters $\lambda$ can be calculated without directly forming the Hessian, since it can be shown to be equivalent to the gradient with respect to the mean parameters $\mu = [m, \text{mm}^\top + \mathbf{P}]$ [25]. The natural parameter update, with learning rate $\beta$, becomes,

$$
\lambda \leftarrow \lambda + \beta \frac{\partial \mathcal{L}}{\partial \mu},
$$

To update the approximation posterior, $\lambda$ can be simply transformed to the moment parameterisation $[m, \mathbf{P}]$. A table of mappings between the various parametrisations is given in App. G.

CVI and the Approximate Likelihood Khan and Lin [31] show that when the prior and approximate posterior are conjugate (as in the GP case), further elegant properties of exponential family distributions mean that Eq. (6) is equivalent to a two step Bayesian update:

$$
\tilde{\lambda} \leftarrow (1 - \beta) \bar{\lambda} + \beta \frac{\partial \mathbb{E}_{q(f)} \left[ \log p(Y \mid f) \right]}{\partial \mu}, \quad \lambda \leftarrow \eta + \tilde{\lambda},
$$

where $\eta$ are the natural parameters of the prior and $\bar{\lambda}$ are the natural parameters of the likelihood contribution. Letting $g(\cdot) = \frac{\partial \mathbb{E}_{q(f)} \left[ \log p(Y \mid f) \right]}{\partial \mu}$, the gradients can be computed in terms of the mean and covariance via the chain rule: $g(\mu) = \left[ g(m) - 2g(\mathbf{P}) m, g(\mathbf{P}) \right]$. Eq. (7) shows that, since the prior parameters $\eta$ are known, natural gradient variational inference is completely characterised by updates to an approximate likelihood, which we denote $\mathcal{N}(\tilde{Y} \mid f, \tilde{V})$, parameterised by covariance $\tilde{V} = (-2\tilde{\lambda}^{(2)})^{-1}$ and mean $\tilde{Y} = \tilde{V} \tilde{\lambda}^{(1)}$ (see App. A). The approximate posterior then has the form,

$$
q(f) = \frac{N(\tilde{Y} \mid f, \tilde{V}) p(f)}{\int N(\tilde{Y} \mid f, \tilde{V}) p(f) df}.
$$

Computing $q(f)$ amounts to performing conjugate GP regression with the model prior and the approximation likelihood. This approach is called conjugate-computation variational inference (CVI, [31]). To re-emphasise that the CVI updates compute the exact same quantity as Eq. (6), we provide an alternative derivation in App. H by directly applying the chain rule to Eq. (6).
3 Spatio-Temporal Variational Gaussian Processes

In this section we introduce a spatio-temporal VGP that has linear complexity with respect to time whilst obtaining the identical variational posterior as the standard VGP. We will then go on to derive this method’s sparse variant, which gives the same posterior as SVGP when the inducing points are set similarly (i.e., on a spatio-temporal grid), but is capable of scaling to much larger values of $M$.

3.1 The Spatio-Temporal VGP ELBO

We first derive our proposed spatio-temporal VGP ELBO. We do this by exploiting the form of the approximate posterior after a natural gradient step in order to write the ELBO as a sum of three terms, each of which can be efficiently computed through filtering and smoothing. As shown in Sec. 2.2, after a natural gradient step, the approximate posterior $q(f) \propto N(\tilde{Y} \mid f, \tilde{V}) p(f)$ decomposes as a Bayesian update applied to the model prior with an approximate likelihood. Following Chang et al. [11] we substitute Eq. (8) into the VGP ELBO:

$$\mathcal{L}_{\text{VGP}} = \mathbb{E}_{q(f)} \left[ \log \frac{p(Y \mid f) p(f)}{q(f)} \right] = \mathbb{E}_{q(f)} \left[ \log \frac{p(Y \mid f) p(f)}{N(\tilde{Y} \mid f, \tilde{V}) p(f)} d\mu \right]$$

$$= \sum_{n=1}^{N_t} \sum_{k=1}^{N_s} \mathbb{E}_{q(f_{n,k})} \left[ \log p(Y_{n,k} \mid f_{n,k}) \right] - \mathbb{E}_{q(f)} \left[ \log N(\tilde{Y} \mid f, \tilde{V}) \right] + \log p(f) \left[ N(\tilde{Y} \mid f, \tilde{V}) \right].$$

The first term is the expected log likelihood, the second is the expected log approximate likelihood, and the final term is the log marginal likelihood of the approximation posterior, $\log p(\tilde{Y}) = \log \mathbb{E}_{p(f)} \left[ N(\tilde{Y} \mid f, \tilde{V}) \right]$. Naively evaluating $\mathcal{L}_{\text{VGP}}$ requires $O(N^3)$ computation for both the update to $q(f)$ and the marginal likelihood. We now show how to compute this with linear scaling in $N_t$.

We observe that after a natural gradient update, $\tilde{V}$, the approximate likelihood covariance, has the same form as the gradient $g(P)$ because, as seen in Eq. (7), $X$ is only additively updated by $g(\mu)$. Since the expected likelihood, $\mathbb{E}_{q(f)} [\log p(Y \mid f)]$, factorises across observations, $g(P)$ is diagonal and hence so is $\tilde{V}$. The approximate likelihood therefore factorises in the same way as the true one:

$$\log N(\tilde{Y} \mid f, \tilde{V}) = \sum_{n=1}^{N_t} \sum_{k=1}^{N_s} \log N(\tilde{Y}_{n,k} \mid f_{n,k}, \tilde{V}_{n,k}).$$

We now turn our attention to computing the posterior and the marginal likelihood. Recall that if the kernel is separable between time and space, $\kappa(t, s, t', s') = \kappa_t(t, t') \kappa_s(s, s')$, then the GP prior can be re-written as Eq. (3). This separability property further results in the state-space model matrices having a convenient Kronecker structure,

$$\tilde{f}(t_{n+1}) = \left[ \mathbf{I}_{N_s} \otimes A^{(t)}_n \right] \tilde{f}(t_n) + q_n, \quad \tilde{Y}_n \mid \tilde{f}(t_n) \sim p(\tilde{Y}_n \mid \tilde{H}(t_n),$$

where $q_n \sim N(0, K^{(g)}_{ss} \otimes Q_n^{(t)})$ and $H = \mathbf{I}_{N_s} \otimes H(t)$. Here $A^{(t)}_n \in \mathbb{R}^{d_t \times d_t}$, $Q_n^{(t)} \in \mathbb{R}^{d_t \times d_t}$, and $H(t) \in \mathbb{R}^{1 \times d_t}$ are the transition matrix, process noise covariance, and measurement model of the SDE (see Sec. 2.1) induced by the kernel $\kappa_t(\cdot, \cdot)$, respectively.

Because the GP prior is Markov and the approximate likelihood factorises across time, the approximate GP posterior is also Markov [45]. Hence marginals $q(f_n)$ can be computed through linear filtering and smoothing applied to Eq. (11). Furthermore, the marginal likelihood of a linear Gaussian state-space model, $p(\tilde{Y}) = p(\tilde{Y}_1) \prod_{n=2}^{N_t} p(\tilde{Y}_n \mid \tilde{Y}_{1:n-1})$, can be computed sequentially by running the forward filter, since $p(\tilde{Y}_n \mid \tilde{Y}_{1:n-1}) = \int p(\tilde{Y}_n \mid \tilde{H}(t_n)) p(\tilde{f}(t_n) \mid \tilde{Y}_{1:n-1}) df(t_n)$, where $p(\tilde{f}(t_n) \mid \tilde{Y}_{1:n-1})$ is the predictive filtering distribution. By combining all of the above properties we can now write the ELBO as,

$$\mathcal{L}_{\text{ST-VGP}} = \sum_{n=1}^{N_t} \sum_{k=1}^{N_s} \mathbb{E}_{q(f_{n,k})} \left[ \log p(Y_{n,k} \mid f_{n,k}) \right] - \sum_{n=1}^{N_t} \sum_{k=1}^{N_s} \mathbb{E}_{q(f_{n,k})} \left[ \log N(\tilde{Y}_{n,k} \mid f_{n,k}, \tilde{V}_{n,k}) \right]$$

$$+ \sum_{n=1}^{N_t} \log \mathbb{E}_{p(f(t_n) \mid \tilde{Y}_{1:n-1})} \left[ N(\tilde{Y}_n \mid \tilde{H}(t_n), \tilde{V}_n) \right].$$

(12)
We now introduce spatial inducing points, $Z$. As discussed in Sec. 2.2, a natural gradient
Algorithm 1

**Spatio-temporal sparse VGP**

| Input: Data: $\{X, Y\}$, Initial params: $\{\tilde{Y}, \tilde{V}\}$, Learning rates: $\{\beta, \rho\}$ | while ELBO ($E$) not converged do |
| $\triangleright$ CVI natural gradient step: $q(u), \ell = \text{Alg. 2}(\tilde{Y}, \tilde{V})$ |
| $E = \mathbb{E}_{q(u)}[\mathbb{E}_{p(f|u)}[\log p(Y|f)]]$ |
| $\tilde{\lambda} = (1-\beta)\lambda + \beta \frac{\partial E}{\partial \mu}$ |
| $\tilde{V} = (-2\tilde{\lambda}^{(2)})^{-1}, \ Y = \tilde{V}\tilde{\lambda}^{(1)}$ |

| $\triangleright$ Hyperparameter gradient step: $q(u), \ell = \text{Alg. 2}(\tilde{Y}, \tilde{V})$ |
| $E = \mathbb{E}_{q(u)}[\mathbb{E}_{p(f|u)}[\log p(Y|f)]]$ |
| $\tilde{\ell} = E - E_{q(u)}[\log N(Y|u, V)] + \ell \triangleright ELBO$ |
| $\theta = \theta + \rho \frac{\partial \tilde{\ell}}{\partial \theta}$ |
| end while |

This ELBO can be computed with linear scaling in $N_t$: $O(N_t N_d^3 d^3)$. We now show that the natural gradient step for updating the parameters of $N(Y|f, \tilde{V})$ can be computed with the same complexity.

3.2 Efficient Natural Gradient Updates

As discussed in Sec. 2.2, a natural gradient update to the posterior, $q(f) \propto p(f) N(\bar{Y} | f, \bar{V})$, has superior convergence properties to gradient descent, and is completely characterised by an update to the approximate likelihood, $N(\bar{Y} | f, \bar{V})$, whose mean and covariance are the free parameters of the model, and implicitly define the same variational parameters as VGP. Since the likelihood factorises across the data points, these updates only require computation of the marginal distribution $q(f_{n,k})$ to obtain $\mathbb{E}_{q(f_{n,k})}[\log p(f_{n,k}|\bar{f}, \bar{Y})]$ and its gradients.

As we have shown, computation of the marginal posterior amounts to smoothing over the state, $\bar{f} \sim N(\bar{f} | \bar{m}, \bar{P})$, with the model in Eq. (11). The time marginals are given by applying the measurement model to the state: $q(f_n) = N(f_n | m_n = H\bar{m}_n, P_n = HP_nH^T)$ after which $q(f_{n,k}) = \int q(f_n) \, df_n \neq k$ can then be obtained by integrating out the other spatial points. Given the marginal, Eq. (7) can be used to give the new likelihood parameters $\bar{Y}$ and $\bar{V}$. The full learning algorithm iterates this process alternately with a hyperparameter update via gradient descent applied to the ELBO, Eq. (12), and has computational complexity $O(N_t N_d^3 d^3)$. We call this method the spatio-temporal variational GP (ST-VGP).

3.3 Spatial Sparsity: from $O(N_t N_d^3 d^3)$ to $O(N_t M_s^3 d^3)$

We now introduce spatial inducing points, $Z_s$, in order to reduce the effective dimensionality of the state-space model. Whilst we maintain the same notation for consistency, it should be noted that the sparse model no longer requires the data to be on a spatio-temporal grid, only that the inducing points are. In this case, letting $q(u) = N(u | m^{(u)}, P^{(u)})$ be the sparse variational posterior, the marginal $q(f_n) = N(f | m_n, P_n)$ only depends on $m_n^{(u)}, P_n^{(u)}$ due to the conditional independence property for separable kernels discussed in Sec. 2.1. We compute the posterior $q(u)$ via filtering and smoothing over the state $\bar{f}(t)$ in a similar way to ST-VGP by setting,

$$A_n = I_{M_s} \otimes A_n^{(t)}, \quad Q_n = K_{Z_s Z_n}^{(a)} \otimes Q_n^{(t)}, \quad H = I_{M_s} \otimes H^{(t)}. \quad (13)$$

Alg. 2 gives the smoothing algorithm. However, the natural gradient update, Eq. (7), now becomes,

$$\tilde{\lambda} \leftarrow (1-\beta)\lambda + \beta \frac{\partial E_{q(u)}[\mathbb{E}_{p(f|u)}[\log p(Y|f)]]}{\partial H^{(u)}}, \quad (14)$$

which results in $\tilde{\lambda}^{(2)}$, and hence also $\tilde{V}$, being a dense matrix (i.e., $\tilde{V}$ is block-diagonal) due to the conditional mapping, $p(f_n | u_n)$. Therefore the approximate likelihood for the sparse model factorises across time, but not space (see App. J for details): $\log N(\bar{Y} | u, \bar{V}) = \sum_{n=1}^{N_t} \log N(\bar{Y}_n | u_n, \bar{V}_n)$. 

This ELBO can be computed with linear scaling in $N_t$: $O(N_t N_d^3 d^3)$. We now show that the natural gradient step for updating the parameters of $N(Y | f, \tilde{V})$ can be computed with the same complexity.

**Algorithm 2**

**Spatio-temporal smoothing**

| Input: Likelihood: $\{\tilde{Y}, \tilde{V}\}$, Space prior: $\{K_{Z_s Z_n}^{(a)}\}$, Time prior: $\{A^{(t)}, Q^{(t)}, H^{(t)}\}$ |
| $\triangleright$ Construct model matrices: $A_n = I_{M_s} \otimes A_n^{(t)}$, $Q_n = K_{Z_s Z_n}^{(a)} \otimes Q_n^{(t)}$, $H = I_{M_s} \otimes H^{(t)}$ |
| $\triangleright$ Filtering and smoothing: |
| if using parallel filter / smoother then $q(u), \ell = \text{Alg. 4}(\tilde{Y}, \tilde{V}, A, Q, H)$ |
| else $q(u), \ell = \text{Alg. 3}(\tilde{Y}, \tilde{V}, A, Q, H)$ |
| end if |
| $\triangleright$ Return posterior marginals and log likelihood: return $q(u), \ell$ |
where the final term is given by the forward filter.

The Spatio-Temporal Sparse VGP ELBO Adding inducing points in space is equivalent to placing
the inducing points on a spatio-temporal grid (i.e., inducing points exist at all time steps), and hence
the variational objective directly follows from \( \mathcal{L}_{\text{SVGP}} \) using a similar argument to Sec. 3.1:

\[
\mathcal{L}_{\text{ST-SVGP}} = \mathbb{E}_{q(f, u)} \left[ \log \frac{p(Y \mid f) p(f \mid \eta, \alpha) \int N(\bar{Y} \mid u, \bar{V}) p(u) \, du}{N(\bar{Y} \mid u, \bar{V}) \, p(f \mid \eta, \alpha)} \right]
\]

\[
= \sum_{n=1}^{N_t} \sum_{k=1}^{N_x} \mathbb{E}_{q(u_n)} \left[ \mathbb{E}_{p(f_{n,k} \mid u_n)} \left[ \log p(Y_{n,k} \mid f_{n,k}) \right] \right] - \sum_{n=1}^{N_t} \mathbb{E}_{q(u_n)} \left[ \log N(\bar{Y}_n \mid u_n, \bar{V}_n) \right]
\]

\[
+ \sum_{n=1}^{N_t} \log \mathbb{E}_{p(f(t_n) \mid Y_{1:n-1})} \left[ N(\bar{Y}_n \mid H f(t_n), \bar{V}_n) \right], \tag{15}
\]

where the final term is given by the forward filter.

Efficient Natural Gradient Updates The marginal required to compute the ELBO and natural
gradient, \( q(f_{n,k}) = \int \int p(f \mid u) q(u) \, du \, df_{\neq n,k} = \int \int p(f \mid u_n) q(u_n) \, du_n \, df_{\neq n,k} \), is the predictive
distribution at input \( X_{n,k} \) from the posterior \( q(u) \). Because the inducing points have only been placed
in space, this can be simplified through the Kronecker structure given by the state-space model. As
shown in App. 1, the marginal mean and covariance are,

\[
m_{n,k} = K^{(s)}_s Z_s K^{(-)}_s Z_s m_{n}^{(u)};
\]

\[
P_{n,k} = K^{(s)}_s Z_s K^{(-)}_s Z_s P_{n}^{(u)} K^{(s)}_s Z_s K^{(-)}_s Z_s + K^{(l)}_s X_{n,k} X_{n,k} \left( K^{(s)}_s K^{(-)}_s - K^{(s)}_s K^{(-)}_s K^{(s)}_s K^{(-)}_s \right), \tag{16}
\]

where \( m_{n}^{(u)} = H m_{n}, P_{n}^{(u)} = H P_{n} H^T \) are given by filtering and smoothing. By combining
the above properties we see that all the terms required for the natural gradient updates and hyperparameter
learning can be computed efficiently in \( O(N_t M^3 d^3) \). We call this approach the spatio-temporal
sparse variational GP (ST-SVGP). The full algorithm is given in Alg. 1.

4 Further Improving the Temporal and Spatial Scaling

We now propose two approaches to further improve the computational properties of ST-VGP and
ST-SVGP. First, we show how parallel filtering and smoothing can be used for non-conjugate GP
inference, which results in a theoretical span complexity of \( O(\log N_t d^2) \). We then present a spatial
mean-field approximation, which can be used independently, or in combination with sparsity.

4.1 Parallel Bayesian Filtering and Smoothing

The associative scan algorithm [9] uses a divide-and-conquer approach combined with parallelisation
to convert \( N \) sequential associative operations into \( \log N \) sequential steps (for an operator \( * \), associativity
implies \( (a * b) * c = a * (b * c) \)). This algorithm has been made applicable to conjugate Markov
We examine the scalability and performance of ST-VGP and its variants. Throughout, we use a Matérn-\(\nu\\)/2 kernel and optimise the hyperparameters by maximising the ELBO using Adam [32].

4.2 Spatial Mean-Field Approximation

We reconsider the state space model for the spatio-temporal GP derived in Sec. 2.1, which has process noise \(q_n \sim \mathcal{N}(0, \mathbf{K}_{ss}^{(n)} \otimes \mathbf{Q}_n^{(t)})\). This Kronecker structure implies \(N_s\) independent processes are linearly mixed using spatial covariance \(\mathbf{K}_{ss}^{(n)}\). The linearity of this operation makes it possible to reformulate the model to include the mixing as part of the measurement, rather than the prior:

\[
\tilde{f}(t_{n+1}) = \mathbf{A}_n \tilde{f}(t_n) + q_n, \quad \mathbf{Y}_n \mid \tilde{f}(t_n) \sim p(\mathbf{Y}_n \mid \mathbf{C}_{ss}^{(n)} \otimes \mathbf{H}^{(t)} \tilde{f}(t_n)),
\]

where \(q_n \sim \mathcal{N}(0, \mathbf{I}_{N_s} \otimes \mathbf{Q}_n^{(t)})\) and \(\mathbf{C}_{ss}^{(n)}\) is the Cholesky factor of \(\mathbf{K}_{ss}^{(n)}\) (see App. F for the derivation).

This has the benefit that now both \(\mathbf{A}_n\) and \(\mathbf{Q}_n = \mathbf{I}_{N_s} \otimes \mathbf{Q}_n^{(t)}\), are block diagonal, such that under the prior the latent processes are fully independent. This enables a mean-field assumption between the \(N_s\) latent posterior processes: \(q(\tilde{f}(t)) \approx \prod_{k=1}^{N_s} q(\tilde{f}_k(t))\), where \(\tilde{f}_k(t)\) is the \(d_t\)-dimensional state corresponding to spatial point \(\mathbf{s}_k\). This approximation enforces block-diagonal structure in the state covariance, such that matrix operations acting on the full state can be avoided. Dependence between the latent processes is still captured via the measurement model (likelihood), and our experiments show that this approach still approximates the true posterior well (see Sec. 5 and App. K.6) whilst providing significant computational gains when \(N_s\) is large.

5 Experiments

We examine the scalability and performance of ST-VGP and its variants. Throughout, we use a Matérn-\(\nu\)/2 kernel and optimise the hyperparameters by maximising the ELBO using Adam [32].
We run SKI with 10 ρ. We use learning rates of with land (N = 30 AIR) because they can afford more inducing points for the same computational budget.

Table 1: NYC-CRIME (small) results. ST-SVGP = SVGP when Z is fixed.

<table>
<thead>
<tr>
<th>TRAIN Z</th>
<th>MODEL</th>
<th>RMSE</th>
<th>NLPD</th>
</tr>
</thead>
<tbody>
<tr>
<td>×</td>
<td>ST-SVGP</td>
<td>3.02 ± 0.13</td>
<td>1.72 ± 0.04</td>
</tr>
<tr>
<td>✓</td>
<td>ST-SVGP</td>
<td>2.79 ± 0.15</td>
<td>1.64 ± 0.04</td>
</tr>
<tr>
<td>✓</td>
<td>SVGP</td>
<td>2.94 ± 0.12</td>
<td>1.65 ± 0.05</td>
</tr>
</tbody>
</table>

We use 5-fold cross-validation (i.e., 80–20 train-test split), train for 500 iterations (except for AIR where we train for 300) and report RMSE, negative log predictive density (NLPD, see App. K.1) and average per-iteration training times on CPU and GPU. When using a GPU, the parallel filter and smoother are used.

**Synthetic Experiment** We construct 7 toy datasets with rich temporal structure but smooth spatial structure (see App. K.2) and varying size: Nt = 10, 82, 155, 227, 300, 500, 1000, and construct a 500×100 grid that serves as a test set for all cases. As the dataset size increases we expect the predictive performance of all methods to improve at the expense of run time. We compare against SKI and SVGP (see Sec. 1.1). Fig. 2a shows that whilst SVGP becomes infeasible for more than 300 time steps, the ST-SVGP variants scale linearly with time and are faster than SKI (except for the very small datasets, in which the model compile time in JAX dominates). In App. K we show the test performance of all models.

**NYC-CRIME – Count Dataset** We model crime numbers across New York City, USA (NYC), using daily complaint data from [1]. Crime data has seasonal trends and is spatially dependent. Accurate modelling can lead to more efficient allocation of police resources [20, 3]. We first consider a small subset of the data to highlight when our methods exactly recover SVGP. We bin the data from 1st to 10th of January 2014 (Nt = 10) into a spatial grid of size 30×30 and drop cells that do not intersect with land (Na = 447, N = 4470). We run ST-SVGP and SVGP with inducing points initialised to the same locations. We plot the training ELBO in Fig. 2b and performance in Table 1. For fixed inducing points, both models have the same training curve and provide the same predictions (up to the same locations. We plot the training ELBO in Fig. 2b and performance in Table 1. For fixed inducing points, both models have the same training curve and provide the same predictions (up to numerical differences). Optimising the inducing points improves both methods. A comparable inference method for non-conjugate likelihoods has not yet been developed for SKI.

We next consider observations from January to July 2014, with daily binning (Nt = 182) and the same spatial grid (Na = 447, N = 81,354). We run ST-SVGP and its mean-field variant (MF-ST-SVGP) with 30 spatial inducing points (equivalent to SVGP with M = 30×182 = 5460). Table 2 shows that our methods outperform SVGP (with M = 1500, M = 3000 and batch sizes 1500, 3000 respectively) because they can afford more inducing points for the same computational budget.

**Regression: AIR-QUALITY** Finally, we model PM10 (µg/m3) air quality across London, UK. The measurements exhibit periodic fluctuations and highly irregular behaviour due to events like weather and traffic jams. Using hourly data from the London air quality network [29] between January 2019 and April 2019 (Nt = 2159), we drop sensors that are not within the London boundaries or have more than 40% of missing data (Na = 72, N = 155,448). We run ST-SVGP and MF-ST-SVGP with 30 inducing points in space (equivalent to SVGP with M = 30×2159 = 64,770 inducing points). To ensure the run times are comparable on both CPU and GPU, we run SVGP with 2000, 2500, 5000, and 8000 inducing points with mini-batch sizes of 600, 800, 2000, and 3000 respectively. We run SKI with Nt temporal inducing points and 6 inducing points in each spatial dimension.

Table 3: AIR-QUALITY regression. ST-SVGP fits the fast-varying structure well, whereas SVGP smooths the data. Average run time and standard deviation in seconds shown for a single training step. ST-SVGP and MF-ST-SVGP use 30 spatial inducing points, equivalent to SVGP with 30×2159 = 64,770 inducing points. Number of inducing points chosen to make run time comparable.
Our methods significantly outperform the SVGP baselines because they can afford considerably more inducing points. As shown in Fig. 3 the SVGP drastically smooths the data, whereas ST-SVGP fits the short-term structure well. The mean-field approach is significantly more efficient, especially when using the parallel algorithm, but we do observe a slight reduction in prediction quality.

6 Conclusions and Discussion

We have shown that variational inference and spatio-temporal filtering can be combined in a principled manner, introducing an approach to GP inference for spatio-temporal data that maintains the benefits of variational GPs, whilst exhibiting favourable scaling properties. Our experiments confirm that ST-SVGP outperforms baseline methods because the effective number of inducing points grows with the temporal horizon, without introducing a significant computational burden. Crucially, this leads to improved predictive performance, because fast varying temporal information is captured by the model. We demonstrated how to apply parallel filtering and smoothing in the non-conjugate GP case, but our empirical analysis identified a maximum state dimension of around \( d \approx 50 \), after which the sub-linear temporal scaling is lost. However, our proposed spatial mean-field approach alleviates this issue somewhat, making the combined algorithm extremely efficient even when both the number of time steps and spatial points are large. The resemblance of our framework to the VGP approach suggests many potential extensions, such as multi-task models [4] and deep GPs [16]. We provide JAX code for all methods at https://github.com/AaltoML/spatio-temporal-GPs.

6.1 Limitations and Societal Impact

We believe our work takes an important step towards allowing sophisticated GP models to be run on both resource constrained CPU machines and powerful GPUs, greatly expanding the usability of such models whilst also reducing unnecessary consumption of resources. However, when using predictions from such methods, the limitations of the model assumptions and potential inaccuracies when using approximate inference should be kept in mind, especially in cases such as crime rate monitoring where actions based on biased or incorrect predictions can have harmful societal consequences.

Acknowledgments and Disclosure of Funding

This work was supported by the Academy of Finland Flagship programme: Finnish Center for Artificial Intelligence (FCAI). Parts of this project were done while OH was visiting Finland as part of the FCAI–Turing initiative, supported by HIIT/FCAI. OH acknowledges funding from The Alan Turing Institute PhD fellowship programme. AS and WJW acknowledge funding from the Academy of Finland (grant numbers 324345 and 339730). NAL contributed under the NVIDIA AI Technology Center (NVAITC) Finland program. TD acknowledges support from EPSRC (EP/T004134/1), UKRI Turing AI Fellowship (EP/V02678X/1), and the Lloyd’s Register Foundation programme on Data Centric Engineering. We acknowledge the computational resources provided by the Aalto Science-IT project and CSC – IT Center for Science, Finland.

References


