Actually Sparse Variational Gaussian Processes

Harry Jake Cunningham[1] **Daniel Augusto de Souza**[1] **So Takao**[1]
Mark van der Wilk[2] **Marc Peter Deisenroth**[1]
University College London[1], Imperial College London[2]

Abstract

Gaussian processes (GPs) are typically criticised for their unfavourable scaling in both computational and memory requirements. For large datasets, sparse GPs reduce these demands by conditioning on a small set of inducing variables designed to summarise the data. In practice however, for large datasets requiring many inducing variables, such as low-lengthscale spatial data, even sparse GPs can become computationally expensive, limited by the number of inducing variables one can use. In this work, we propose a new class of inter-domain variational GP, constructed by projecting a GP onto a set of compactly supported B-spline basis functions. The key benefit of our approach is that the compact support of the B-spline basis functions admits the use of sparse linear algebra to significantly speed up matrix operations and drastically reduce the memory footprint. This allows us to very efficiently model fast-varying spatial phenomena with tens of thousands of inducing variables, where previous approaches failed.

1 INTRODUCTION

Gaussian processes (GPs) (Rasmussen and Williams, 2006) provide a rich prior over functions. Their non-parametric form, gold-standard uncertainty estimates and robustness to overfitting have made them common place in geostatistics (Oliver and Webster, 1990), epidemiology (Bhatt et al., 2017), spatio-temporal modelling (Blangiardo et al., 2013; Wikle et al., 2019), robotics and control (Deisenroth and Rasmussen, 2011) and Bayesian optimisation (Osborne et al., 2009). However, GPs scale infamously as $\mathcal{O}(N^3)$ in computational complexity and $\mathcal{O}(N^2)$ in memory, where N is the size of the training dataset, making them unfeasible for use with large datasets. To overcome this limitation, there exist a number of different approximate inference techniques, including sparse approximations (Snelson and Ghahramani, 2006; Quinonero-Candela and Rasmussen, 2005; Titsias, 2009), state-space methods (Hartikainen and Särkkä, 2010; Särkkä et al., 2013; Hamelijnck et al., 2021) and local-expert models (Tresp, 2000a,b; Rasmussen and Ghahramani, 2001; Deisenroth and Ng, 2015; Cohen et al., 2020). In particular, sparse GP approximations have been developed to reduce the cubic complexity of inference by introducing a set of inducing variables. Sparse approaches summarise the training data by a set of $M \ll N$ pseudo-data, effectively reducing the rank of the covariance matrix. Amongst these methods, variational approximations have proved popular in improving GPs for regression (Titsias, 2009), classification (Hensman et al., 2015b), stochastic optimisation (Hensman et al., 2013), inference with non-conjugate likelihoods (Hensman et al., 2015b,a) and hierarchical non-parametric modelling (Damianou and Lawrence, 2013; Salimbeni and Deisenroth, 2017).

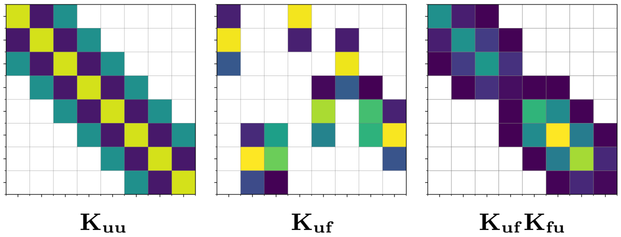


Figure 1: Illustration of the sparse matrix structures induced by our proposed method for 1D regression with a Matérn-3/2 kernel. By constructing inter-domain inducing variables $\mathbf{u}$ as RKHS projections of the GP onto a set of compactly supported B-splines, both the inducing point covariance matrix $\mathbf{K_{uu}}$ and the covariance matrix between the GP f and the inducing variables $\mathbf{K_{uf}}$ become sparse. This admits sparse linear algebra to precompute the sparse matrix product $\mathbf{K_{uf}K_{fu}}$, which is used to compute the ELBO.

Introduced by Titsias (2009), Sparse Variational Gaussian processes (SVGPs) approximate the true GP posterior with an approximate one, conditioned on a set of M induc-

Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206. Copyright 2023 by the author(s).

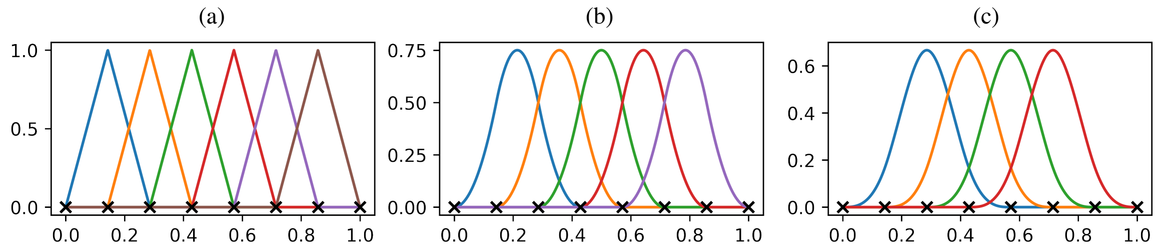


Figure 2: (a) 1st-order B-spline basis (b) 2nd-order B-spline basis (c) 3rd-order B-spline basis. For the same set of knots, the support of the B-splines increases in width with increasing order. This has the effect that each B-spline basis function has intersecting support with an increasing number of basis functions as the order increases.

ing variables. The approximate posterior is then learnt by minimising the Kullback-Leibler (KL) divergence between the approximate and true posterior, allowing us to learn the variational parameters and hyperparameters jointly via gradient descent. The resulting approximation scales as $\mathcal{O}(NM^2 + M^3)$ in computational complexity and $\mathcal{O}(NM)$ in memory. However, these low-rank approximations are practically limited to $\approx 10,000$ inducing points, which can be insufficient for complex datasets where a large number of inducing points are required to cover the input space. This limitation is especially apparent in long time-series or spatial datasets with intrinsically low lengthscales, where traditional low-rank approximations based on small sets of localised pseudo-datapoints fail to capture fast variations in the data (Pleiss et al., 2020; Wu et al., 2022).

To alleviate some of these problems, inter-domain GPs (Lázaro-Gredilla and Figueiras-Vidal, 2009; van der Wilk et al., 2020) generalise the idea of inducing variables by transforming the GP to a different domain by means of a linear operator, which admits more expressive features and/or computationally efficient linear algebra. Variational Fourier Features (VFFs) (Hensman et al., 2017), constructs inter-domain inducing variables by projecting the GP onto a Fourier basis. This results in inducing variables that span the width of the domain and therefore describe global variations in the data. By the orthogonality of the Fourier basis, the inducing variables are also almost independent, producing computationally efficient block-diagonal covariance matrices. In one dimension, this can be exploited to reduce the computational complexity to $\mathcal{O}(M^3)$ after an initial one-off pre-computation of $\mathcal{O}(NM^2)$. However, since the Fourier basis functions are global, whilst computationally efficient, they are inefficient at modelling low-lengthscale data. Indeed, VFF typically requires more inducing variables for an equivalent accuracy than standard sparse GP regression for $d \geq 2$ (Hensman et al., 2017).

Variational Inducing Spherical Harmonics (VISH) by Dutordoir et al. (2020) remedied some of the problems faced by VFF by first projecting the data onto a D-dimensional unit hypersphere and then using a basis of spherical harmonics as inter-domain inducing features. As the basis functions are orthogonal, VISH reduces the cost of matrix inversion to $\mathcal{O}(M)$ and the total cost of inference to $\mathcal{O}(NM^2)$. However, by projecting data onto the hypersphere and performing sparse GP regression on the transformed space, VISH is unable to use covariance functions which use the Euclidean distance between data points. This makes VISH sub-optimal for naturally Euclidean spatial data.

In this work, we propose a new inter-domain approach that scales GPs to complex datasets that require a very large number of inducing variables. Specifically, we define a new inter-domain approximation by projecting the GP onto a basis of compactly supported B-splines. Due to the local support of the B-spline basis functions, the covariance between inducing variables yields sparse band-diagonal covariance matrices, admitting highly efficient sparse linear algebra at a complexity that scales linearly with the number of inducing variables. In contrast to both VFF and VISH, which use basis functions with global support, our choice of basis also incites sparse structure in the covariance between inducing variables and the GP itself. Our results show that our method is particularly well suited to spatial data with high-frequency variations, which necessitate a large number of inducing variables. By using computationally cheap, locally supported inducing variables, we can cover the domain with many basis functions that are able to successfully capture local variations.

2 BACKGROUND

A Gaussian process is a collection of random variables, any finite number of which is jointly Gaussian distributed. A GP is fully characterised by its mean $\mu(\cdot)$ and covariance function $k(\cdot,\cdot)$ (Rasmussen and Williams, 2006). Given a training dataset $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^{N}$ of N noisy observations $y_n \in \mathbb{R}$ and corresponding inputs $\mathbf{x}_n \in \mathbb{R}^D$, and observation model $y_n = f(\mathbf{x}_n) + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma^2)$, we construct a GP regression problem by placing a zero-mean GP prior on the latent function $f \sim \mathcal{GP}(0, k(\cdot,\cdot))$. The posterior distribution $p(f|\mathbf{y}) \sim \mathcal{GP}(\mu(\cdot), \Sigma(\cdot,\cdot))$ is a GP with

$$\begin{aligned} \mu(\cdot) &= \mathbf{k}_{\mathbf{f}}^T(\cdot)\mathbf{K}_{\mathbf{yy}}^{-1}\mathbf{y}, \\ \Sigma(\cdot,\cdot) &= k(\cdot,\cdot) - \mathbf{k}_{\mathbf{f}}^T(\cdot)\mathbf{K}_{\mathbf{yy}}^{-1}\mathbf{k}_{\mathbf{f}}(\cdot), \end{aligned} \tag{1}$$

where $\mathbf{k_f}(\cdot) = [k(\mathbf{x}_n, \cdot)]_{n=1}^N$, $\mathbf{K_{yy}} = \mathbf{K_{ff}} + \sigma^2\mathbf{I}$ and $\mathbf{K_{ff}} = [k(\mathbf{x}_i, \mathbf{x}_j)]_{i,j=1}^N$.

To train the GP we maximise the log-marginal likelihood $\log p(\mathbf{y}) = \log \int p(\mathbf{y}|\mathbf{f})p(\mathbf{f})\mathrm{d}\mathbf{f}$. In the case of a Gaussian likelihood, this takes the explicit form

$$\log p(\mathbf{y}) = -\frac{1}{2}\mathbf{y}^\top \mathbf{K}_{\mathbf{yy}}^{-1}\mathbf{y} - \frac{1}{2}\log|\mathbf{K_{yy}}| - \frac{n}{2}\log 2\pi. \quad (2)$$

Training the GP scales in $\mathcal{O}(N^3)$ due to computing the matrix inverse and determinant in (2). Moreover, when using gradient-based optimisation to tune the hyperparameters, (2) must be computed at every iteration. Predictions using (1) require $\mathcal{O}(N^2)$ computations, assuming $\mathbf{K}_{\mathbf{yy}}^{-1}$ (or its Cholesky factorisation) has been cached, e.g., after the training procedure. In terms of memory, GP predictions require $\mathcal{O}(N^2)$ to store the Cholesky factor of $\mathbf{K_{yy}}$. The computational and memory demands therefore make GPs prohibitively expensive for datasets with more than $\approx 10,000$ datapoints.

2.1 Sparse Variational Gaussian Processes

Variational inference provides an elegant method to approximate the true posterior $p(f|\mathbf{y})$ of a GP with a variational distribution $q(f)$, rather than approximating the model itself. Sparse variational Gaussian processes (SVGPs) introduced by Titsias (2009) leverage inducing points coupled with variational inference to construct a low-rank approximation to the posterior. SVGP consists of introducing a (small) set of inducing variables $\mathbf{u} = \{f(\mathbf{z}_m)\}_{m=1}^M$ defined at a set of inducing point locations $Z = \{\mathbf{z}_m\}_{m=1}^M$. Placing a Gaussian distribution over the inducing variables $q(\mathbf{u}) = \mathcal{N}(\mathbf{m}, \mathbf{S})$, the approximate posterior

$$q(f) = \int p(f|\mathbf{u})q(\mathbf{u})\mathrm{d}\mathbf{u} = \mathcal{GP}(\mu(\cdot), \Sigma(\cdot,\cdot)) \quad (3)$$

is obtained by marginalising out the inducing variables. The approximate posterior (3) is defined in terms of the variational parameters $\mathbf{m} \in \mathbb{R}^M$ and $\mathbf{S} \in \mathbb{R}^{M\times M}$, where, due to the conjugacy between $p(f|\mathbf{u})$ and $q(\mathbf{u})$,

$$\mu(\cdot) = \mathbf{k}_\mathbf{u}^T(\cdot)\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{m}, \quad (4)$$

$$\Sigma(\cdot,\cdot) = k(\cdot,\cdot) + \mathbf{k}_\mathbf{u}^T(\cdot)\mathbf{K}_{\mathbf{uu}}^{-1}(\mathbf{S} - \mathbf{K_{uu}})\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{k_u}(\cdot). \quad (5)$$

Here $\mathbf{k_u}(\cdot) = [\mathrm{cov}(u_m, f(\cdot))]_{m=1}^M = [k(\mathbf{z}_m, \cdot)]_{m=1}^M$ and $\mathbf{K_{uu}} = [\mathrm{cov}(u_i, u_j)]_{i,j=1}^M = [k(\mathbf{z}_i, \mathbf{z}_j)]_{i,j=1}^M$.

The variational parameters $\mathbf{m}$ and $\mathbf{S}$ are optimised by minimising the KL divergence between the true and approximate posterior KL $[q(f) \,\|\, p(f|\mathbf{y})]$. In practice, this is made tractable by maximising the evidence lower bound (ELBO)

$$\mathcal{L}_{\mathrm{ELBO}} = \sum_{n=1}^N \mathbb{E}_{q(f_n)}[\log p(y_n|f_n)] - \mathrm{KL}\left[q(\mathbf{u}) \,\|\, p(\mathbf{u})\right], \quad (6)$$

which provides a lower bound to the log-marginal likelihood $\log p(\mathbf{y}) \geq \mathcal{L}_{\mathrm{ELBO}}$, and whose gap is precisely the KL divergence that we are minimising. Normally, the hyperparameters of the model are optimised jointly with the variational parameters, by maximising the ELBO.

For a Gaussian likelihood, the moments of the optimal distribution $\hat{q}(\mathbf{u}) = \mathcal{N}(\hat{\mathbf{m}}, \hat{\mathbf{\Sigma}})$ can be computed exactly as

$$\hat{\mathbf{m}} = \sigma^{-2}\hat{\mathbf{\Sigma}}\mathbf{K_{uf}}\mathbf{y}, \quad (7)$$

$$\hat{\mathbf{\Sigma}} = \mathbf{K_{uu}}\left[\mathbf{K_{uu}} + \sigma^{-2}\mathbf{K_{uf}}\mathbf{K_{fu}}\right]^{-1}\mathbf{K_{uu}}. \quad (8)$$

The corresponding optimal ELBO is given by

$$\begin{aligned}\mathcal{L}_{\mathrm{ELBO}} &= \log\mathcal{N}\left(\mathbf{y}|\mathbf{0}, \mathbf{K_{fu}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K_{uf}} + \sigma_n^2\mathbf{I}\right) \\ &\quad - \frac{1}{2}\sigma_n^{-2}\mathrm{tr}\left(\mathbf{K_{ff}} - \mathbf{K_{fu}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K_{uf}}\right),\end{aligned} \quad (9)$$

where $\mathbf{K_{uf}} = [k(\mathbf{z}_m, \mathbf{x}_n)]_{m,n=1}^{M,N}$. SVGPs thus reduce the computational cost of training to $\mathcal{O}(NM^2 + M^3)$ per evaluation of the ELBO. Hensman et al. (2013) showed that the ELBO in (6) is also amenable to stochastic optimisation, further reducing the computational complexity to $\mathcal{O}(N_bM^2 + M^3)$ per iteration by using minibatches. SVGPs require $\mathcal{O}(N_bM + M^2)$ memory to store $\mathbf{K_{fu}}$ and the dense Cholesky factor of $\mathbf{K_{uu}}$.

The use of a low-rank approximation does have certain tradeoffs, however. Whilst small M speeds up computation, the choice of M is also essential to ensuring a certain quality of approximation (Burt et al., 2019). Using a small number of inducing points becomes particularly troublesome for data with inherently short lengthscales, which commonly occurs when working with spatial data. In this case, the SVGP will collapse quickly to the prior mean and variance when not in the immediate vicinity of an inducing input.

2.2 Variational Fourier Features (VFF)

Inter-domain GPs (Alvarez and Lawrence, 2008; Lázaro-Gredilla and Figueiras-Vidal, 2009; van der Wilk et al., 2020) generalise the idea of inducing variables by instead conditioning on a linear transformation $\mathcal{L}_m$ of the GP $\mathbf{u} = [\mathcal{L}_m f(\cdot)]_{m=1}^M$. By choosing $\mathcal{L}_m$ to be a convolution of $f(\cdot)$ with respect to a Dirac delta function centred at the inducing points $\mathbf{z}_m$, we can recover the standard inducing point approximation. However, by choosing different linear operators, such as projections (Hensman et al., 2017; Dutordoir et al., 2020) or general convolutions (van der Wilk et al., 2017), we can construct more informative features, without changing the sparse variational inference scheme.

VFF (Hensman et al., 2017) is an inter-domain variational GP approximation that constructs inducing features as a Matérn RKHS projection of the GP onto a set of Fourier basis functions $u_m = \langle f, \phi_m\rangle_\mathcal{H}, m = 1, \ldots, M$, where $\langle\cdot,\cdot\rangle_\mathcal{H}$ denotes the Matérn RKHS inner product, and $\phi_0(x) = 1$, $\phi_{2i-1}(x) = \cos(\omega_i x)$, $\phi_{2i} = \sin(\omega_i x)$ are the Fourier basis functions. This results in the matrices

$$\mathbf{K_{uu}} = [\langle\phi_i, \phi_j\rangle_\mathcal{H}]_{i,j=1}^M, \quad \mathbf{K_{uf}} = [\phi_m(x_n)]_{m,n=1}^{M,N} \quad (10)$$

where, due to the reproducing property, the cross-covariance matrix $\mathbf{K_{uf}}$, which is equivalent to evaluating the Fourier basis, is independent of kernel hyperparameters. This leads to several computational benefits: (i) we can precompute $\mathbf{K_{uf}}$, as it remains constant throughout hyper-parameter training via the ELBO (9), (ii) due to the orthogonality of the Fourier basis, $\mathbf{K_{uu}}$ is the sum of a block-diagonal matrix plus low-rank matrices, e.g., in the case of a 1D Matérn-1/2 kernel,

$$\mathbf{K_{uu}} = \mathrm{diag}(\boldsymbol{\alpha}) + \boldsymbol{\beta}\boldsymbol{\beta}^\top \tag{11}$$

for some $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}^M$, where the vector $\boldsymbol{\beta}$ is sparse. This structure can be exploited to significantly reduce the computational complexity for training and prediction when compared to standard sparse GP methods. However, VFF has two main flaws:

- VFF generalises poorly to higher dimensions due to the use of a Kronecker product basis. This construction of a high-dimensional basis not only scales exponentially in the number of dimensions, it is also inefficient in terms of captured variance (Dutordoir et al., 2020): Multiplying together basis functions of increasing frequency causes the prior variance to decay rapidly, resulting in large numbers of redundant features and the down-weighting of important low-frequency ones. Thus for $D \geq 2$, VFF typically requires more inducing variables than SGPR, making it memory inefficient.
- Whilst $\mathbf{K_{uu}}$ has a computationally efficient structure, $\mathbf{K_{uf}}$ is still a dense matrix. In the special case when the likelihood is Gaussian, we still require to compute a dense Cholesky factor of the $M \times M$ matrix $\mathbf{K_{uu}} + \sigma^{-2}\mathbf{K_{uf}}\mathbf{K_{fu}}$ (see (8)), which costs $\mathcal{O}(M^3)$. The same problem persists for VISH.

In order to address these issues, in the next section we will consider defining inter-domain inducing variables as the projection of the GP onto a set of compactly supported basis functions, drastically reducing memory requirements and improving computational efficiency, enabling us to use large numbers of inducing points.

3 B-SPLINE INDUCING FEATURES

In this section, we introduce B-spline inducing features and propose Actually Sparse Variational Gaussian Processes (AS-VGPs). The core idea is to use the concept of RKHS projections as in VFF, except to project a GP onto a set of compactly supported *B-spline basis functions* instead of the Fourier basis functions. Unlike in VFF, the resulting inducing features $\{u_m\}_{m=1}^M$ are localised by the nature of their compact support, see Figure 2, such that $\mathbf{K_{uu}}$, $\mathbf{K_{uf}}$ and $\mathbf{K_{uf}}\mathbf{K_{fu}}$ are *all* sparse matrices (see Figure 1). These sparse covariance structures allow us to gain substantial computational benefits.

3.1 B-Spline Inducing Features

B-spline basis functions of order k are a set of compactly supported piece-wise polynomial functions of degree k. Their shape is controlled by an increasing sequence of knots $V = \{v_m\}_{m=0}^M \in \mathbb{R}$ that partition the domain into M sub-intervals. We denote the m-th B-spline basis function of order k by $B_{m,k}(x)$ (See Appendix E for expressions). Since a k-th order B-spline has compact support over only $k+1$ sub-intervals, it has intersecting support with at most $k+1$ other B-spline basis functions (see Figure 2).

We define the *B-spline inducing features* as the RKHS projection $u_m = \langle f, \phi_m(\cdot)\rangle_{\mathcal{H}}$ onto the B-spline basis, where $\phi_m(x) = B_{m,k}(x)$. Under this choice, the covariance between the inducing features u_m and the GP f is given by

$$[\mathbf{K_{uf}}]_{m,n} = \mathrm{Cov}[u_m, f(x_n)] = \langle k(x_n, \cdot), \phi_m(\cdot)\rangle_{\mathcal{H}} \tag{12}$$
$$= \phi_m(x_n) = B_{m,k}(x_n) \tag{13}$$

and reduces to a simple evaluation of the B-spline basis at the training inputs. Note that $B_{m,k}(x) \neq 0$ if and only if $x \in [v_m, v_{m+k+1}]$ and therefore $\mathbf{K_{uf}}$ is sparse with at most $M(k+1)$ non-zero entries. As with VFF, $\mathbf{K_{uf}}$ is also independent of the kernel hyperparameters, meaning it remains constant throughout training and can be precomputed. Next, the covariance between the inducing features is given by

$$[\mathbf{K_{uu}}]_{m,m'} = \mathrm{Cov}[u_m, u_{m'}] = \langle \phi_m, \phi_{m'}\rangle_{\mathcal{H}}, \tag{14}$$

which is only non-zero when ϕ_m and $\phi_{m'}$ have intersecting support. This produces sparse band-diagonal $\mathbf{K_{uu}}$ matrices with bandwidth equal to $k+1$. Since the B-spline basis functions are piecewise polynomials, we can evaluate the inner product in closed form, allowing for efficient computation during training and testing.

Remark 1. Strictly speaking, the notation $\langle f, \phi_m(\cdot)\rangle_{\mathcal{H}}$ is ill-defined, as samples of f are almost surely not elements of $\mathcal{H}$ Kanagawa et al. (2018). In order to make rigorous sense of this, we use the machinery of generalised Gaussian fields, which we discuss in Appendix D.

3.2 Sparse Linear Algebra

In this section, we will initially restrict our analysis to GPs with one-dimensional inputs and extend this later in Section 3.4 to higher dimensions. Using our proposed spline inducing features, we have the following desirable properties that we can leverage in the key computations (8)–(9):

Property 1: For the Matérn-$\nu/2$ class of kernels, $\mathbf{K_{uu}}$ is a band-diagonal matrix with bandwidth equal to *at least* $\nu/2 + 3/2$.

This is due to the fact that, in order to be a valid projection, the B-spline basis functions must belong to the same Matérn RKHS. As stated by Kanagawa et al. (2018), the RKHS generated by the Matérn-$\nu/2$ kernel $k(\cdot, \cdot)$ is norm-equivalent to

Table 1: Complexity of sparse variational GPs for evaluating the ELBO (9) in 1D regression settings with a Gaussian likelihood. N: number of datapoints; M: number of inducing points; k: bandwidth of the covariance matrix; N_b: size of the mini-batch in stochastic variational inference. For both VFF and VISH we quote the complexity required for exact SGPR.

Algorithm	Pre-computation	Computational complexity	Storage
SGPR (Titsias, 2009)	✗	$\mathcal{O}(NM^2 + M^3)$	$\mathcal{O}(NM)$
SVGP (Hensman et al., 2013)	✗	$\mathcal{O}(N_bM^2 + M^3)$	$\mathcal{O}(M^2 + N_bM)$
VFF (Hensman et al., 2017)	$\mathcal{O}(NM^2)$	$\mathcal{O}(NM^2 + M^3)$	$\mathcal{O}(M^2 + NM)$
VISH (Dutordoir et al., 2020)	$\mathcal{O}(NM^2)$	$\mathcal{O}(NM^2)$	$\mathcal{O}(M^2 + NM)$
AS-VGP (Ours)	$\boldsymbol{\mathcal{O}(N)}$	$\boldsymbol{\mathcal{O}((k+1)^2M)}$	$\boldsymbol{\mathcal{O}((k+1)M)}$

the Sobolev space $\mathcal{H}^{\nu/2+1/2}$. Given their polynomial form, we can check that B-splines of order k are C^{k-1}-smooth and moreover k-times weakly differentiable (see Appendix E). Since the B-splines are compactly supported, so are their (weak) derivatives; therefore, the (weak) derivatives are all square-integrable. Thus, they belong to the Sobolev space $\mathcal{H}^k$. As a result, for the Matérn-$\nu/2$ kernel, we choose to project onto B-splines of order $k = \nu/2 + 1/2$, giving us a $\mathbf{K_{uu}}$ matrix with bandwidth $k + 1 = \nu/2 + 3/2$.

Property 2: The matrix product $\mathbf{K_{uf}K_{fu}}$ is a band-diagonal matrix with bandwidth at most equal to that of $\mathbf{K_{uu}}$.

To see this, from (13) we have

$$[\mathbf{K_{uf}K_{fu}}]_{ij} = \sum_{n=1}^{N}[\mathbf{K_{uf}}]_{in}[\mathbf{K_{uf}}]_{jn} \tag{15}$$

$$= \sum_{n=1}^{N} B_{i,k}(x_n)B_{j,k}(x_n). \tag{16}$$

By the properties of B-splines, $B_{i,k}(x_n)B_{j,k}(x_n) \neq 0$ if and only if $x_n \in \mathcal{I}_{ij}$, where $\mathcal{I}_{ij} = [v_i, v_{i+k+1}] \cap [v_j, v_{j+k+1}]$ is the intersection of the supports of the two B-splines $B_{i,k}$ and $B_{j,k}$. However, we know that the supports are intersecting if and only if $|i - j| < k + 1$. Hence, when $|i - j| \geq k + 1$, no data point can be contained in $\mathcal{I}_{ij}$ since it is the empty set, giving us $B_{i,k}(x_n)B_{j,k}(x_n) = 0$ for all $n = 1, \ldots, N$ and therefore $[\mathbf{K_{uf}K_{fu}}]_{ij} = 0$ from (16). This implies that the matrix $\mathbf{K_{uf}K_{fu}}$ has bandwidth at most equal to $k + 1$.

Using these two properties, we can construct an inter-domain variational method that can leverage sparse linear algebra to speed up inference and significantly save on memory footprint. We discuss this next.

3.3 Actually Sparse Variational Gaussian Processes

We propose Actually Sparse Variational Gaussian Processes (AS-VGP) as inter-domain variational GPs that use B-Spline inducing variables. For one-dimensional GPs, our method has several computational advantages:

- $\mathbf{K_{uf}}$ is very sparse with typically 1% of its entries being non-zero. This allows us to store it as a sparse tensor, resulting in 2 orders of magnitude memory saving.
- By Properties (1)–(2), the sum $\mathbf{K_{uu}} + \sigma^{-1}\mathbf{K_{uf}K_{fu}}$ in (8) is band-diagonal and its inverse can be computed at a cost of $\mathcal{O}(M(k+1)^2)$; its memory footprint is $\mathcal{O}(M(k+1))$.
- Using the banded operators from Durrande et al. (2019), we compute $\mathrm{tr}(\mathbf{K_{fu}K_{uu}^{-1}K_{uf}}) = \mathrm{tr}(\mathbf{K_{uu}^{-1}K_{uf}K_{fu}})$ in (9) without having to instantiate a dense matrix, reducing the memory footprint to $\mathcal{O}(M(k+1))$.

Overall, this reduces the pre-computation cost for computing the sparse matrix multiplication $\mathbf{K_{uf}K_{fu}}$ to *linear* in the number of training datapoints. The resulting matrix can be cached for later use with a memory footprint of $\mathcal{O}((k+1)M)$, owing to its banded structure (Property 2). Further, the per-iteration computational cost and memory footprint of computing the ELBO (9) and its gradients is also *linear* in the number of inducing variables, required to take the (sparse) Cholesky decomposition of a banded matrix (Durrande et al., 2019).

Further, using the banded operators introduced by Durrande et al. (2019), given a banded Cholesky factor of $\mathbf{K_{uu}}$, we can also compute only the band elements of its inverse at a cost of $\mathcal{O}(M(k+1)^2)$. Given that $\mathbf{K_{uf}K_{fu}}$ is a banded matrix (Property 2), we compute the trace term in (9) by computing only the bands of the matrix product $\mathbf{K_{uu}^{-1}K_{uf}K_{fu}}$, with the computational cost $\mathcal{O}(M(k+1)^2)$, thereby avoiding *ever* instantiating a dense matrix.

We compare the compute and memory costs of various sparse GP inference algorithms in Table 1. This highlights the linear scaling in both memory and computational complexity with inducing points of the proposed AS-VGP. Compared to both VFF and VISH, AS-VGP is the only method that scales linearly in both computational complexity and storage, enabling it to be used with tens or hundreds of thousands of inducing variables, significantly more than both VFF and VISH.

Table 2: Predictive mean squared errors (MSEs), negative log predictive densities (NLPDs) and wall-clock time in seconds with one standard deviation based on 5 random splits for a number of UCI regression datasets. All models use a Matérn-3/2 kernel and L-BFGS optimiser.

			MSE ($\times 10^{-1}$)			NLPD		
Dataset	N	M	SGPR	VFF	AS-VGP	SGPR	VFF	AS-VGP
Air Quality	9k	500	6.43 ± 0.04	6.64 ± 0.04	6.68 ± 0.04	1.24 ± 0.00	1.25 ± 0.00	1.25 ± 0.00
Synthetic	10k	50	0.40 ± 0.00	0.39 ± 0.00	0.39 ± 0.00	-0.16 ± 0.00	-0.15 ± 0.00	-0.15 ± 0.00
Rainfall	43k	700	0.48 ± 0.00	0.83 ± 0.00	0.84 ± 0.00	0.10 ± 0.00	0.25 ± 0.00	0.29 ± 0.00
Traffic	48k	300	9.96 ± 0.01	10.01 ± 0.01	10.02 ± 0.01	1.42 ± 0.00	1.42 ± 0.00	1.42 ± 0.00

3.4 Extensions to Higher Dimensions

To extend AS-VGP to higher dimensions, we employ a similar strategy to VFF, by constructing either the additive or separable kernel. In the separable case, we have

$$k(\mathbf{x}, \mathbf{x}') = \prod_{d=1}^{D} k_d(x_d, x'_d), \tag{17}$$

where $k_d(\cdot, \cdot)$ for $d = 1, \dots, D$ are one-dimensional kernels. By choosing the basis functions to be a tensor product of M one-dimensional B-Splines, that is,

$$\boldsymbol{\phi}(\mathbf{x}) = \bigotimes_{d=1}^{D} [B^{(d)}_{m,k}(x_d)]_{m=1}^{M} \in \mathbb{R}^{M^D}, \tag{18}$$

we get the matrices

$$\mathbf{K_{uf}} = [\boldsymbol{\phi}(\mathbf{x}_n)]_{n=1}^{N}, \quad \mathbf{K_{uu}} = \bigotimes_{d=1}^{D} \mathbf{K}^{(d)}_{\mathbf{uu}}, \tag{19}$$

computed using (13)–(14), where $\mathbf{K}^{(d)}_{\mathbf{uu}}$ for $d = 1, \dots, D$ denotes the matrix (14) corresponding to the 1D case and $\{\mathbf{x}_n\}_{n=1}^{N}$ are the training inputs. Note that some of the structures present for one-dimensional inputs are also present in the Kronecker formulation, namely: (i) $\mathbf{K_{uu}}$ is a block-banded matrix with bandwidth $\approx kM^{D-1}$ whose Cholesky factorisation can be computed in $\mathcal{O}(kM^{D-1})$, (ii) $\mathbf{K_{uf}}\mathbf{K_{fu}}$ is also a band-diagonal matrix with bandwidth $\approx kM^{D-1}$. For low-dimensional problems, the large number of basis functions (18) provides a rich covering of the input space. However, this is unsuitable for large D due to the exponential scaling in the number of input dimensions.

For the additive case, we construct D-dimensional kernels as the sum of D one-dimensional kernels, i.e.,

$$k(\mathbf{x}, \mathbf{x}) = \sum_{d=1}^{D} k_d(x_d, x'_d). \tag{20}$$

This results in a band-diagonal $\mathbf{K_{uu}}$ matrix with bandwidth equal to the one-dimensional equivalent. However, the product $\mathbf{K_{uf}}\mathbf{K_{fu}}$ is no longer sparse and hence inference using a Gaussian likelihood and pre-computation requires an $\mathcal{O}(DM^3)$ Cholesky factorisation.

4 EXPERIMENTS

In the following, we evaluate AS-VGP on a number of regression tasks. We highlight the following properties of our method: 1) AS-VGP significantly reduces the memory requirements of sparse variational GPs without sacrificing on performance. 2) AS-VGP is extremely fast and scalable (training on 2 million 1D data points and 1000 inducing points in under 6 seconds). 3) AS-VGP is able to perform closed-form optimal variational inference when other methods have to use stochastic optimisation instead. 4) AS-VGP is not limited to low-dimensional problems and improves upon VFF when using an additive structure. 5) AS-VGP is particularly suited to modelling fast-varying spatial datasets.

4.1 One-Dimensional Regression

Regression Benchmarks. The purpose of this experiment is to assess the empirical performance and computational benefits of AS-VGP in comparison with SVGP and VFF on medium-sized datasets. We use three UCI benchmarks and a synthetic dataset to compare the predictive performance of AS-VGP with SVGP and VFF. For the synthetic dataset, we generated a periodic function to compare our locally supported B-Spline basis with VFF, which uses a naturally periodic basis.

For each dataset, we randomly sample 90% of the data for training and 10% for testing, repeating this five times to calculate the mean and standard deviation, of the predictive performance (MSE) and uncertainty quantification (NLPD). When using AS-VGP, we normalise the inputs to be between $[0, M]$, where M is the number of inducing points to ensure the spacing between knots is equal to 1, to avoid numerical issues caused by large gradients when computing the inner-product between basis functions. All models are trained using the L-BFGS optimiser; for VFF and AS-VGP, we precompute the matrix product $\mathbf{K_{uf}}\mathbf{K_{fu}}$. We use the Matérn-3/2 kernel for each experiment.

The results in Table 2 demonstrate that AS-VGP is comparative in performance to VFF on every dataset, whilst being less memory intensive. This highlights how our locally supported basis functions offer benefits in both complexity and

Table 3: Predictive mean squared errors (MSEs), negative log predictive densities (NLPDs) and wall-clock time in seconds with one standard deviation based on 5 random splits of the household electric power consumption dataset containing $2,049,279$ data points. The number of inducing variables used is given by M.

Method	$M = 1000$	$M = 5000$	$M = 10,000$	$M = 20,000$	$M = 30,000$
AS-VGP (MSE $\times 10^{-1}$)	**8.65± 0.00**	**6.55± 0.01**	**4.53 ± 0.00**	**3.41± 0.01**	**2.90± 0.01**
SVGP (MSE $\times 10^{-1}$)	9.00 ± 0.01	/	/	/	/
AS-VGP (NLPD)	**1.34 ± 0.00**	**1.20 ± 0.00**	**1.01 ± 0.00**	**0.86 ± 0.00**	**0.77 ± 0.00**
SVGP (NLPD)	1.37 ± 0.00	/	/	/	/
AS-VGP (Time in s)	**5.51 ± 0.10**	**14.4 ± 0.23**	**24.5 ± 0.35**	**46.3 ± 0.35**	**75.0 ± 1.70**
SVGP (Time in s)	188 ± 1.18	/	/	/	/

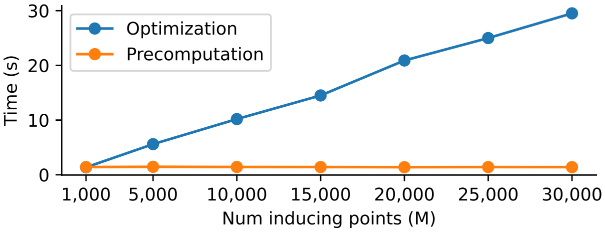


Figure 3: Illustration of the linear scaling in computation of the ELBO and independence on computing the product $\mathbf{K_{uf}K_{fu}}$ w.r.t. the number of inducing points.

memory over their globally supported counterparts, while retaining comparable performance. We note that SGPR performs slightly better than both VFF and AS-VGP, but at a higher computational complexity and without the ability for pre-computation.

Large-Scale Regression. In this example, we illustrate the scalability of our method both in the number of data points and in the number of inducing points, using the household electric power consumption dataset, where $N = 2,049,279$. We opt to use the entire dataset, which uses a one-minute sampling rate over a period of four years, as an example of data with a very low lengthscale. This necessitates a large number of inducing variables and tests the model's ability to scale accordingly. We repeat each experiment five times by randomly sampling 95% of the data for training and use the remaining 5% for evaluation. For each experiment, we use the Matérn-3/2 kernel.

Results are displayed in Table 3.We were unable to use either VFF or VISH in this experiment as we were unable to precompute $\mathbf{K_{uf}K_{fu}}$ due to $\mathbf{K_{uf}}$ not fitting on GPU memory. For SVGP, we used minibatching to reduce the reliance on memory. However, we also ran into issues when using $M \geq 1000$ requiring us to use a very ($N_b = 100$) small batchsize given the size of the dataset and the model became computationally unfeasible for $M \geq 5,000$. In contrast, for AS-VGP, memory was not an issue, and we were able to efficiently scale the number of inducing points.

As highlighted in Table 3, our method was more than two orders of magnitude faster than SVGP, fitting a GP with $1,000$ inducing points and over 2 million datapoints in under 6 seconds. We also observe that the time taken for each AS-VGP experiment follows a linear trend shown in Figure 4.1 as predicted (see Table 1). AS-VGP was also more accurate than SVGP both in predictive performance (MSE) and uncertainty quantification (NLPD), and showed an increase in performance as more inducing variables were added. Firstly, this is indicative of optimal closed-form variational inference being a better approximation to the true posterior than stochastic variational inference. Secondly, this emphasises how the B-spline basis is able to accurately represent local variance in the data and motivates using a large number of inducing points when the lengthscale is very small.

4.2 Additive Regression

In this experiment we show that AS-VGP is not limited to low-dimensional problems, but can scale to high dimensions using an additive structure and use large numbers of inducing points.

The airline dataset is a common GP benchmark, consisting of flight details for every commercial flight in the USA from 2008. The task is to predict the amount of delay y given eight different covariates (route distance, airtime, aircraft, age, etc.). We follow the exact same setup as from Hensman et al. (2013) using an additive Matérn-3/2 GP and evaluate the performance on four datasets of size $10K$, $100K$, $1,000K$ and $5,929,413$ (complete dataset) by subsampling the original data. For each dataset, we perform 10 splits, using two thirds of the data for training and a third for testing. We report the mean and standard deviation of the MSE and NLPD in Table 4. For AS-VGP, we normalise the inputs to be between $[0, M]$, where M is the number of inducing points to ensure the spacing between knots is equal to 1.

We compare our method using both $M = 30$ (240 in total) basis functions and $M = 200$ (1600 in total) basis functions per dimension. Table 4 shows that by adding more basis functions, we can improve upon VFF and VISH in terms

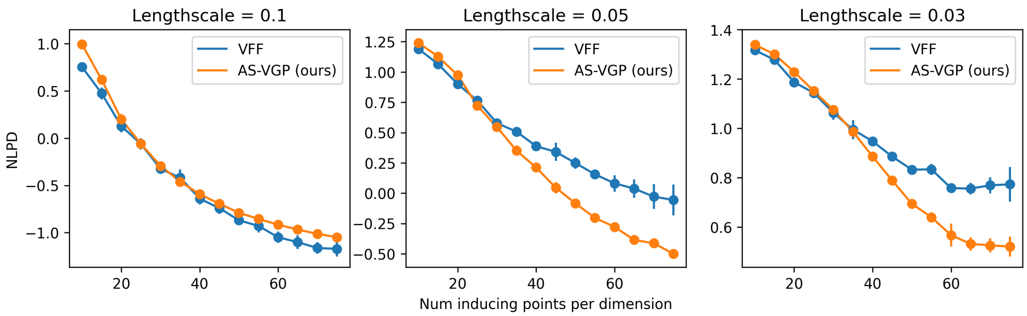


Figure 4: Mean NLPD for AS-VGP and VFF for increasing numbers of inducing points. The data is obtained by sampling a GP with Matérn-3/2 kernel with decreasing lengthscale. The mean NLPD for each model is computed with known parameters and by averaging over five separate samples. The error bars show one standard deviation.

Table 4: Predictive mean squared errors (MSEs), negative log predictive densities (NLPDs) and wall-clock time in seconds with one standard deviation based on 5 random splits for a number of UCI regression datasets. All models use a Matérn-3/2 kernel and the L-BFGS optimiser for training. All models show comparable performance.

		$N = 10,000$		$N = 100,000$		$N = 1,000,000$		$N = 5,929,413$	
Model	M	MSE	NLPD	MSE	NLPD	MSE	NLPD	MSE	NLPD
VISH	610	0.90±0.16	1.33±0.09	0.81±0.05	1.27±0.03	0.83±0.03	1.28±0.01	0.83±0.06	1.27±0.00
VFF	30/dim	0.89±0.15	1.36±0.09	0.82±0.05	1.32±0.03	0.83±0.01	1.34±0.01	0.83±0.00	1.32±0.00
AS-VGP	30/dim	0.95±0.17	1.39±0.09	0.84±0.05	1.33±0.03	0.84±0.01	1.33±0.01	0.83±0.00	1.33±0.00
AS-VGP	200/dim	0.91±0.16	1.37±0.09	0.82±0.05	1.32±0.03	0.83±0.01	1.32±0.01	0.82±0.00	1.32±0.00

of MSE on the larger datasets. We can also scale to larger numbers of inducing points. While VFF uses 240 inducing points in total, we use 1600 in our largest experiment, while remaining computationally efficient since pre-computation is independent of the number of inducing points (and linear in the number of training datapoints; see Table 1).

4.3 Synthetic Spatial Data

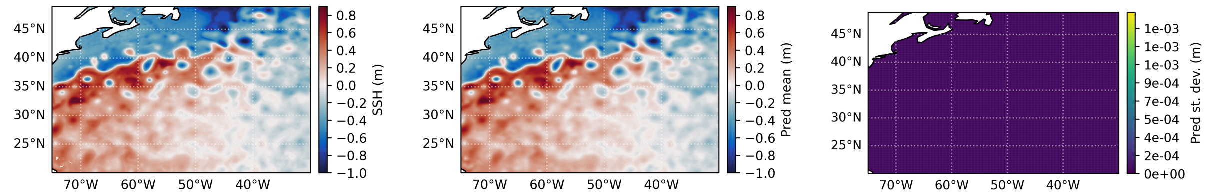
In the following, we demonstrate the effectiveness of AS-VGP on synthetic spatial data with an inherently low lengthscale. To simulate high fidelity spatial data with fast variations, we sample from a 2D GP with a kernel constructed as the product of two 1D Matérn-3/2 kernels. We generate data by sampling the GP five times each for three different lengthscales: 0.1, 0.05 and 0.03. Fixing the lengthscales of AS-VGP and VFF to match the generated data, we then compute the NLPD, for different numbers of inducing points.

Figure 4 shows that AS-VGP captures the variance in the data better than VFF as the lengthscale is reduced. This is in part the fault of the product basis in VFF, which produces features with very small variance, becoming more pronounced with features of higher-frequency (Dutordoir et al. (2020)). However, it also promotes the use of compactly supported basis functions which, unlike the Fourier basis that describe the process across the entire domain, act locally and therefore are more effective at modelling local variations in the data.

4.4 Real-World Spatial Data

In this experiment, we test AS-VGP on a very large spatial regression problem. For this we use the eNATL60 ocean model of sea surface height (SSH) over the North Atlantic at $1/60°$ grid resolution as a real-world example of an extremely large low-lengthscale spatial regression. We perform a typical regridding problem by interpolating the model data defined on a curvilinear grid onto a regular latitude-longitude grid. We restrict the domain to a $45° \times 30°$ region and randomly select 2 million data points from the model as training observations and $100,000$ points for testing. We then evaluate the trained AS-VGP model on a regular grid at $1/12°$ resolution, equivalent to a 540×360 grid.

We fit AS-VGP on this data using 100 basis function per dimension (10,000 in total) in 109 seconds, 41 seconds for pre-computation and 68 seconds for optimisation, achieving an MSE of 9.3×10^{-4} and NLPD of 2.1 on the test set. Similar to the 1D large-scale regression experiment, we could not use the equivalent VFF or SGPR model as storing $\mathbf{K_{uf}}$ (a $2,000,000 \times 10,000$ matrix), requires 149 GB of memory, which cannot be stored on GPU or CPU memory. In contrast, for AS-VGP, storing $\mathbf{K_{uf}}$ only requires 216 MB of memory. Consequently, AS-VGP can handle both large numbers of datapoints and inducing points without requiring stochastic optimsation, which is unachievable using both SGPR and VFF. Using the trained model, we then predict onto a regular latitude longitude grid at $1/12°$ resolution,



(a) Ground truth.

(b) Predictive mean.

(c) Predictive standard deviation.

Figure 5: Real-world data from the eNATL60 ocean model over the Gulfstream at $1/60°$ grid resolution. (a) Ground truth; (b) Predictive mean and (c) predictive standard deviation for AS-VGP at a regular grid with $1/12°$ resolution. The predictive mean of the AS-VGP and the ground truth are nearly identical while the predictive uncertainty essentially vanishes.

taking 36 seconds. Figure 5 shows that AS-VGP is able to model the small structures present in the SSH, quickly and efficiently, whilst still being able to perform closed-form optimal variational inference where other methods can't.

5 DISCUSSION

For one-dimensional inputs with a Gaussian likelihood, AS-VGP is extremely fast, scalable and lightweight, exploiting the band-diagonal structure of both $\mathbf{K_{uu}}$ and $\mathbf{K_{uf}}\mathbf{K_{fu}}$ to perform pre-computation that scales linearly in the number of datapoints and evaluations of the ELBO that scale linearly in the number of inducing variables. We also show that our method is not limited to one-dimensional inputs, but can scale to higher dimensions using an additive or Kronecker structure. In particular, we show that our method is particularly strong at representing processes with small lengthscales, making it amenable to modelling spatio-temporal data or long time series.

Stochastic variational inference also enables the scaling of GPs to large datasets via mini-batching. In practice, however, when using a Gaussian likelihood and if compute permits, SGPR produces a better approximation to the true posterior than SVGP. Whilst in regular SGPR the cost to compute the ELBO is dependent on N, VFF made it possible to remove this by performing a one-off pre-computation, effectively scaling SGPR to millions of data points. Our work extends VFF even further by reducing the complexity of the pre-computation with respect to the number of datapoints and decoupling it from the number of inducing variables, enabling us to scale to larger N and larger M.

Comparisons to our method can also be made to Structured Kernel Interpolation (SKI) by Wilson and Nickisch (2015), but from a variational perspective. Both SKI and AS-VGP construct inducing variables on dense grids. However, whereas SKI performs explicit interpolation between inducing points, AS-VGP implicitly performs interpolation by instead constructing inducing variables equivalent to evaluating a set of B-spline basis functions.

Whilst, we show good performance in low-dimensional problems, ideally, we would not have to impose a Kronecker structure that scales so badly in dimensionality, but instead project directly onto a set of 2D basis functions. Taking inspiration from the connections with SKI, a better choice of basis might be one defined on the simplex, which offers linear scaling in D when generalising to higher dimensions (Kapoor et al., 2021).

Limitations The main limitation of our approach is the scaling to high dimensions. Unlike VISH, we inherit many of the shortcoming of VFF, including a reliance on tensor products which requires an exponential increase in the number of basis functions with increasing dimensions. However, by decoupling the pre-computation from the number of inducing variables, our method is less affected by exponential scaling than VFF. For low numbers of inducing features M, our method performs worse than VFF. However, we can mitigate this shortcoming by using more inducing features due to the linear scaling in M versus cubic for VFF (see Table 1). Finally, like VFF our method currently only supports the Matérn class of kernels. A future research direction would be to expand the class of kernels that can be decomposed using B-splines, e.g., non-stationary kernels, which could help improve spatial modelling.

In practise we propose to use our method on low-dimensional problems ($D \leq 4$), such as spatial or spatio-temporal data, where our method has shown to be computationally and memory efficient while being able to capture high-frequency variations.

6 CONCLUSION

We introduced a novel inter-domain GP model wherein the inducing features are defined as RKHS projections of the GP onto compactly-supported B-spline basis functions. This results in covariance matrices that are sparse, allowing us to draw entirely on techniques from sparse linear algebra to do GP training and inference and thereby opening the door to GPs with tens of thousand inducing variables. Our experiments demonstrate that we get significant computational speed up and memory savings without sacrificing accuracy.

Acknowledgements

HJC is supported by the UCL Department of Computer Science DTP scholarship.

References

Mauricio Alvarez and Neil D Lawrence. Sparse convolved Gaussian processes for multi-output regression. *Advances in Neural Information Processing Systems*, 2008.

Samir Bhatt, Ewan Cameron, Seth R Flaxman, Daniel J Weiss, David L Smith, and Peter W Gething. Improved prediction accuracy for disease risk mapping using Gaussian process stacked generalization. *Journal of the Royal Society Interface*, Volume 14:20170520, 2017.

Marta Blangiardo, Michela Cameletti, Gianluca Baio, and Håvard Rue. Spatial and spatio-temporal models with R-INLA. *Spatial and Spatio-temporal Epidemiology*, 4: 33–49, 2013.

Viacheslav Borovitskiy, Alexander Terenin, P Mostowsky, and Marc P Deisenroth. Matérn Gaussian processes on Riemannian manifolds. *Advances in Neural Information Processing Systems*, 2020.

David Burt, Carl E Rasmussen, and Mark van der Wilk. Rates of convergence for sparse variational Gaussian process regression. In *International Conference on Machine Learning*, 2019.

Samuel Cohen, Rendani Mbuvha, Tshilidzi Marwala, and Marc P Deisenroth. Healing products of Gaussian process experts. In *International Conference on Machine Learning*, 2020.

Andreas Damianou and Neil D Lawrence. Deep Gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, 2013.

Marc P Deisenroth and Jun Wei Ng. Distributed Gaussian processes. In *International Conference on Machine Learning*, 2015.

Marc P Deisenroth and Carl E Rasmussen. PILCO: A model-based and data-efficient approach to policy search. In *International Conference on Machine Learning*, 2011.

Nicolas Durrande, James Hensman, Magnus Rattray, and Neil D Lawrence. Detecting periodicities with Gaussian processes. *PeerJ Computer Science*, 2:e50, 2016.

Nicolas Durrande, Vincent Adam, Lucas Bordeaux, Stefanos Eleftheriadis, and James Hensman. Banded matrix operators for Gaussian Markov models in the automatic differentiation era. In *International Conference on Artificial Intelligence and Statistics*, 2019.

Vincent Dutordoir, Nicolas Durrande, and James Hensman. Sparse Gaussian processes with spherical harmonic features. In *International Conference on Machine Learning*, 2020.

Oliver Hamelijnck, William Wilkinson, Niki Loppi, Arno Solin, and Theodoros Damoulas. Spatio-temporal variational Gaussian processes. *Advances in Neural Information Processing Systems*, 2021.

Jouni Hartikainen and Simo Särkkä. Kalman filtering and smoothing solutions to temporal Gaussian process regression models. In *International Workshop on Machine Learning for Signal Processing*, 2010.

James Hensman, Nicolò Fusi, and Neil D Lawrence. Gaussian processes for big data. In *International Conference on Uncertainty in Artificial Intelligence*, 2013.

James Hensman, Alexander G Matthews, Maurizio Filippone, and Zoubin Ghahramani. MCMC for variationally sparse Gaussian processes. In *Advances in Neural Information Processing Systems*, 2015a.

James Hensman, Alexander G Matthews, and Zoubin Ghahramani. Scalable variational Gaussian process classification. In *International Conference on Artificial Intelligence and Statistics*, 2015b.

James Hensman, Nicolas Durrande, and Arno Solin. Variational Fourier features for Gaussian processes. *Journal of Machine Learning Research*, 18(1):5537–5588, 2017.

Motonobu Kanagawa, Philipp Hennig, Dino Sejdinovic, and Bharath K Sriperumbudur. Gaussian processes and kernel methods: A review on connections and equivalences. *arXiv preprint arXiv:1807.02582*, 2018.

Sanyam Kapoor, Marc Finzi, Ke A Wang, and Andrew G Wilson. SKIing on simplices: Kernel interpolation on the permutohedral lattice for scalable Gaussian processes. In *International Conference on Machine Learning*, 2021.

Miguel Lázaro-Gredilla and Anibal Figueiras-Vidal. Inter-domain Gaussian processes for sparse inference using inducing features. *Advances in Neural Information Processing Systems*, 2009.

Sergey V Lototsky and Boris L Rozovsky. *Stochastic Partial Differential Equations*. Springer, 2017.

Margaret A Oliver and Richard Webster. Kriging: A method of interpolation for geographical information systems. *International Journal of Geographical Information System*, 4(3):313–332, 1990.

Michael A Osborne, Roman Garnett, and Stephen J Roberts. Gaussian processes for global optimization. In *International Conference on Learning and Intelligence Optimization*, 2009.

Geoff Pleiss, Martin Jankowiak, David Eriksson, Anil Damle, and Jacob Gardner. Fast matrix square roots with applications to Gaussian processes and Bayesian optimization. *Advances in Neural Information Processing Systems*, 2020.

Hartmut Prautzsch, Wolfgang Boehm, and Marco Paluszny. *Bézier and B-spline Techniques*, volume 6. Springer, 2002.

Joaquin Quinonero-Candela and Carl E Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6:1939–1959, 2005.

Carl E Rasmussen and Zoubin Ghahramani. Infinite mixtures of Gaussian process experts. *Advances in Neural Information Processing Systems*, 2001.

Carl E Rasmussen and Christopher KI Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Hugh Salimbeni and Marc P Deisenroth. Doubly stochastic variational inference for deep Gaussian processes. *Advances in Neural Information Processing Systems*, 2017.

Simo Särkkä, Arno Solin, and Jouni Hartikainen. Spatiotemporal learning via infinite-dimensional Bayesian filtering and smoothing: A look at Gaussian process regression through Kalman filtering. *IEEE Signal Processing Magazine*, 30(4):51–61, 2013.

Edward Snelson and Zoubin Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems*, 2006.

Michalis Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *International Conference on Artificial intelligence and Statistics*, 2009.

Volker Tresp. A Bayesian committee machine. *Neural Computation*, 12(11):2719–2741, 2000a.

Volker Tresp. Mixtures of Gaussian processes. *Advances in Neural Information Processing Systems*, 2000b.

Mark van der Wilk, Carl E Rasmussen, and James Hensman. Convolutional Gaussian processes. *Advances in Neural Information Processing Systems*, 2017.

Mark van der Wilk, Vincent Dutordoir, ST John, Artem Artemev, Vincent Adam, and James Hensman. A framework for interdomain and multioutput Gaussian processes. *arXiv preprint arXiv:2003.01115*, 2020.

Christopher K Wikle, Andrew Zammit-Mangion, and Noel Cressie. *Spatio-Temporal Statistics with R*. Chapman and Hall/CRC, 2019.

Andrew G Wilson and Hannes Nickisch. Kernel interpolation for scalable structured Gaussian processes (KISS-GP). In *International Conference on Machine Learning*, 2015.

Luhuan Wu, Geoff Pleiss, and John Cunningham. Variational nearest neighbor Gaussian processes. *arXiv preprint arXiv:2202.01694*, 2022.

A CODE

Code is available at `https://github.com/HJakeCunningham/ASVGP`

B EXPERIMENT DETAILS

The timed experiments (Table 3) were performed using a AMD Ryzen 2920X 12-Core CPU and an NVIDIA GeForce RTX 2080 GPU. Below, we include specific details on the two experiments conducted.

Regression Benchmarks For the synthetic dataset, we generate 10,000 random noisy observations from the test function

$$f(x) = \sin(3\pi x) + 0.3\cos(9\pi x) + \frac{\sin(7\pi x)}{2}.$$

Metrics We use the mean-squared error (MSE) and the negative log-predictive density (NLPD) to evaluate the performance of our model. These are defined as

$$\text{MSE}(\{X_n, y_n\}_{n=1}^N) = \frac{1}{N}\sum_{n=1}^N \|y_n - \mu(X_n)\|^2, \tag{21}$$

$$\text{NLPD}(\{X_n, y_n\}_{n=1}^N) = -\frac{1}{N}\sum_{n=1}^N \log \int p(y_n|f_n)\,\mathcal{N}(f_n|\mu(X_n), \xi(X_n))\,\mathrm{d}f_n, \tag{22}$$

where μ, ξ are the posterior mean and variance, respectively.

C RKHS INNER PRODUCTS

The inner products corresponding to the Matérn-1/2 and Matérn-3/2 RKHS defined over the domain $\mathcal{D} = [a, b]$, as given by Durrande et al. (2016); Hensman et al. (2017), are

$$\langle f, g\rangle_{\mathcal{H}_{k_{1/2}}} = \frac{l}{2\sigma^2}\int_a^b f'g'\mathrm{d}x + \frac{1}{2l\sigma^2}\int_a^b fg\,\mathrm{d}x + \frac{1}{2\sigma^2}[f(a)g(a) + f(b)g(b)], \tag{23}$$

$$\begin{aligned}\langle f, g\rangle_{\mathcal{H}_{k_{3/2}}} = &\frac{l^3}{12\sqrt{3}\sigma^2}\int_a^b f''g''\mathrm{d}x + \frac{l}{2\sqrt{3}\sigma^2}\int_a^b f'g'\mathrm{d}x + \frac{\sqrt{3}}{4l\sigma^2}\int_a^b fg\,\mathrm{d}x \\ &+ \frac{1}{2\sigma^2}[f(a)g(a) + f(b)g(b)] + \frac{l^2}{2\sigma^2}[f'(a)g'(a) + f'(b)g'(b)],\end{aligned} \tag{24}$$

respectively, where l, σ are the lengthscale and amplitude hyperparameters.

When performing RKHS projections, it is important to note that the B-splines must belong to the RKHS defined by our choice of kernel. As stated by Kanagawa et al. (2018), the RKHS generated by the Matérn-$\nu/2$ kernel is norm-equivalent to the Sobolev space $\mathcal{H}^{\nu/2+1/2}$. Due to their piecewise polynomial form, B-splines of order k can be shown to belong to the Sobolev space $\mathcal{H}^k$ (see Section E).

From Section 3.2, to minimise computational complexity, we wish to use B-spline basis functions with minimal bandwidth of the $\mathbf{K_{uu}}$ matrix. As a result, for the Matérn-$\nu/2$ kernel we project onto B-splines of order $\nu/2 + 1/2$.

D GENERALISED GAUSSIAN FIELDS OVER RKHS

In this appendix, we justify our abuse of notation $\langle f, \phi\rangle_{\mathcal{H}}$ when f is a GP whose RKHS is $\mathcal{H}$. To this end, we first introduce the notion of a generalised Gaussian field as follows.

Definition 1 (Lototsky and Rozovsky (2017))**.** Let $\mathcal{H}$ be a Hilbert space and $(\Omega, \mathcal{F}, \mathbb{P})$ a probability space. We say that a function $F : \Omega \times \mathcal{H} \to \mathbb{R}$ is a zero-mean generalised Gaussian field over $\mathcal{H}$ if the random variable $Fh := F(\cdot, h)$ is Gaussian for all $h \in \mathcal{H}$ and the following property holds:

1. $\mathbb{E}[Fh] = 0$ for all $h \in \mathcal{H}$, and
2. $\mathrm{Cov}[Fg, Fh] = \langle g, h \rangle_{\mathcal{H}}$ for all $g, h \in \mathcal{H}$, where $\langle \cdot, \cdot \rangle_{\mathcal{H}} : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$ denotes the inner product on $\mathcal{H}$.

In the special case when $\mathcal{H}$ is an RKHS over a base space X with kernel $k : X \times X \to \mathbb{R}$, we can further identify generalised Gaussian fields $F : \Omega \times \mathcal{H} \to \mathbb{R}$ with a stochastic process $f : \Omega \times X \to \mathbb{R}$ over X by the following relation:

$$f(x) = Fk(\cdot, x), \quad \text{for all} \quad x \in X. \tag{25}$$

Notice that $k(\cdot, x) \in \mathcal{H}$ so the expression on the RHS of (25) makes sense. Indeed, GPs of the Matérn class can be identified with generalised Gaussian field over the Sobolev space Borovitskiy et al. (2020). Hence, for any Matérn-$\nu/2$ GP f and any $\phi \in \mathcal{H} := \mathcal{H}^{\nu/2+1/2}$, we can define the RKHS projection of f onto ϕ by

$$\langle f, \phi \rangle_{\mathcal{H}} := F\phi, \tag{26}$$

where F is the generalised Gaussian field corresponding to the process f.

E IMPLEMENTATION OF B-SPLINES

The m-th B-spline basis function of order k, which we denote by $B_{m,k}(x)$ can be computed according to the Cox-de-Boor recursion formula (Prautzsch et al., 2002)

$$B_{m,0}(x) = \begin{cases} 1, & \text{if } v_m \leq x \leq v_{m+1}, \\ 0, & \text{otherwise}, \end{cases} \tag{27}$$

$$B_{m,k}(x) = \frac{x - v_m}{v_{m+k} - v_m} B_{m,k-1}(x) + \frac{v_{m+k+1} - x}{v_{m+k+1} - v_{m+1}} B_{m+1,k-1}(x). \tag{28}$$

The case $k = 0$ corresponds to a top-hat function $\mathbf{1}_{[v_m, v_{m+1}]}(x)$, with support spanning a single sub-interval. In the case $k = 1$, we have the piecewise linear function

$$B_{m,1}(x) = \begin{cases} \dfrac{x - v_m}{v_{m+1} - v_m}, & \text{for } x \in [v_m, v_{m+1}], \\ \dfrac{v_{m+2} - x}{v_{m+2} - v_{m+1}}, & \text{for } x \in [v_m, v_{m+1}], \\ 0, & \text{otherwise}, \end{cases} \tag{29}$$

which corresponds to the tent map, spanning two sub-intervals (see Figure 2 (a)). In the case $k = 2$, we have the piecewise quadradic function

$$B_{m,2}(x) = \begin{cases} \dfrac{(x - v_m)^2}{(v_{m+2} - v_m)(v_{m+1} - v_m)}, & \text{for } x \in [v_m, v_{m+1}], \\ \dfrac{(x - v_m)(v_{m+2} - x)}{(v_{m+2} - v_m)(v_{m+2} - v_{m+1})} + \dfrac{(v_{m+3} - x)(x - v_{m+1})}{(v_{m+3} - v_{m+1})(v_{m+2} - v_{m+1})}, & \text{for } x \in [v_{m+1}, v_{m+2}], \\ \dfrac{(v_{m+3} - x)^2}{(v_{m+3} - v_{m+1})(v_{m+3} - v_{m+2})}, & \text{for } x \in [v_{m+2}, v_{m+3}], \\ 0, & \text{otherwise}, \end{cases} \tag{30}$$

spanning three sub-intervals (see Figure 2 (b)). Likewise, we can construct a piecewise cubic polynomial corresponding to the case $k = 3$ (Figure 2 (c)).

We claim that the B-spline $B_{m,k}(x)$ is of class C^{k-1} for $k \geq 2$ and is moreover k-times weakly differentiable for $k \geq 1$. To prove the first claim, when $k \geq 2$, one can show that (Prautzsch et al., 2002)

$$\frac{\mathrm{d}B_{m,k}(x)}{\mathrm{d}x} = \frac{k}{v_{m+k} - v_m} B_{m,k-1}(x) - \frac{k}{v_{m+k+1} - v_{m+1}} B_{m+1,k-1}(x). \tag{31}$$

Thus, the first derivative of $B_{m,k}(x)$ is a linear combination of B-splines of order $k-1$, its second derivative is a linear combination of B-splines of order $k-2$, and so on. Now, since B-splines of order $k-i$ are continuous in x if and only if $k-i \geq 1$, we see that the i-th derivative of $B_{m,k}(x)$ is continuous in x provided $i \in \{1, \dots, k-1\}$. This implies that $B_{m,k}(x)$ is of class C^{k-1}.

Next, we show that $B_{m,k}(x)$ is k-times weakly differentiable. In the case $k \geq 2$, by the previous arguments, we know that its $k-1$-th derivative exists and moreover can be expressed as a linear combination of first order B-splines. Hence, all we need to demonstrate is that the weak derivative of first order B-splines exists. This can be shown easily, with

$$D_x^w B_{m,1}(x) = \begin{cases} \dfrac{1}{v_{m+1} - v_m}, & \text{for } x \in [v_m, v_{m+1}], \\ \dfrac{1}{v_{m+1} - v_{m+2}}, & \text{for } x \in [v_m, v_{m+1}], \\ 0, & \text{otherwise}, \end{cases} \tag{32}$$

where D_x^w denotes the weak derivative with respect to x. This also trivially implies the case $k = 1$. Hence, $B_{m,k}(x)$ is k-times weakly differentiable for all $k \geq 1$.