Scalable Exact Inference in Multi-Output Gaussian Processes

Wessel P. Bruinsma[1,2] **Eric Perim**[2] **Will Tebbutt**[1] **J. Scott Hosking**[3]
Arno Solin[4] **Richard E. Turner**[1,5]
[1]University of Cambridge, [2]Invenia Labs, [3]British Antarctic Survey, [4]Aalto University, [5]Microsoft Research

Abstract

Multi-output Gaussian processes (MOGPs) leverage the flexibility and interpretability of GPs while capturing structure across outputs, which is desirable, for example, in spatio-temporal modelling. The key problem with MOGPs is the cubic computational scaling in the number of both inputs (*e.g.*, time points or locations), n, and outputs, p. Current methods reduce this to $O(n^3m^3)$, where $m < p$ is the desired degrees of freedom. This computational cost, however, is still prohibitive in many applications. To address this limitation, we present the Orthogonal Linear Mixing Model (OLMM), an MOGP in which exact inference scales linearly in m: $O(n^3m)$. This advance opens up a wide range of real-world tasks and can be combined with existing GP approximations in a plug-and-play way as demonstrated in the paper. Additionally, the paper organises the existing disparate literature on MOGP models into a simple taxonomy called the Mixing Model Hierarchy (MMH).

1 INTRODUCTION

Application areas such as geostatistics, sensor networks, psychometrics, neuroscience, brain imaging, and finance all utilise multi-output statistical models. A rich, but disparate literature has deployed Gaussian processes (GPs, Rasmussen and Williams, 2006) in such modelling problems, the aim being to leverage their interpretability, modularity, and tractability. Indeed, GPs have been successfully applied in a wide variety of single-output contexts: they can automatically discover structure in signals (Duvenaud, 2014), achieve state-of-the-art performance in regression tasks (Bui et al., 2016), enable data-efficient models in reinforcement learning (Deisenroth and Rasmussen, 2011), and support many applications in probabilistic numerics (Hennig et al., 2015), such as in optimisation (Brochu et al., 2010) and quadrature (Minka, 2000).

One of the first applications of GPs to multiple outputs was in geostatistics (Matheron, 1969). Today, multi-output GP (MOGP) models can be found in various areas, including geostatistics (Wackernagel, 2003), factor analysis (Teh and Seeger, 2005; Yu et al., 2009), dependent or multi-task learning (Boyle and Frean, 2005; Bonilla et al., 2007, 2008; Osborne et al., 2008), latent force models (Álvarez et al., 2009; Álvarez and Lawrence, 2009; Álvarez et al., 2010; Álvarez and Lawrence, 2011), state space modelling (Särkkä et al., 2013), regression networks (Wilson et al., 2012; Nguyen and Bonilla, 2014; Dezfouli et al., 2017), and mixture models (Ulrich et al., 2015; Bruinsma, 2016; Parra and Tobar, 2017; Requeima et al., 2019). Although many of these models are constructed in a similar spirit, they differ in their details and terminology, which makes them difficult to compare and organise.

A key practical problem with existing MOGPs is their computational complexity. For n input points each having p outputs, inference and learning in general MOGPs take $O(n^3p^3)$ time and $O(n^2p^2)$ memory, although these may be alleviated by a wide range of approximations (Candela and Rasmussen, 2005; Titsias, 2009; Lázaro-Gredilla et al., 2010; Hensman et al., 2013; Wilson and Nickisch, 2015; Bui et al., 2017; Cheng and Boots, 2017; Hensman et al., 2018). To mitigate these unfavourable scalings, a particular class of MOGPs, called the Linear Mixing Model (LMM, Sec. 2.2), exploits low-rank structure of their covariance via the matrix inversion and determinant lemmas to reduce the complexity of inference and learning to $O(n^3m^3)$ time and $O(n^2m^2)$ memory, where m is the desired degrees of freedom of the observations, typically much less than p. However, many multi-output regression problems cannot be solved adequately with only a small number of degrees of freedom, making the LMM prohibitively expensive in such cases. For example, consider temperatures measured across the Earth's surface concatenated into a vector. Due to the complexity of the climate, we anticipate behaviour at many spatial length scales, so a model that supposes that this vector lives in a particularly low-dimensional "pancake"[1] will not accurately reflect reality.

We present a unifying view of MOGPs to simplify the complicated web of models in this class. We call this the *Mixing Model Hierarchy* (MMH). Building on the Linear Mixing Model (LMM), which is the class of models at the bottom of the MMH, we introduce the *Orthogonal Linear Mixing Model* (OLMM). The OLMM is a particular LMM in which inference and learning take $O(n^3m)$ time and $O(n^2m)$ memory, without significantly sacrificing expressivity nor requiring any approximations. The OLMM is simple to implement and trivially compatible with existing single-output GP scaling techniques to scale to large numbers of data points n. In particular, for a large class of multi-output problems, *e.g.* many spatio-temporal problems, the time and memory complexity can be further reduced to $O(nm)$, which is linear in *both* the number of data points n and degrees of freedom m. We demonstrate the OLMM in experiments on synthetic and real-world data sets.

2 MULTI-OUTPUT GP MODELS

For tasks with p outputs, multi-output Gaussian processes induce a prior distribution over *vector-valued* functions $f\colon \mathcal{T} \to \mathbb{R}^p$ by requiring that any finite number of function values $f_{p_1}(t_1), \ldots, f_{p_n}(t_n)$ with $(p_i)_{i=1}^n \subseteq \{1,\ldots,p\}$ are multivariate Gaussian distributed. Often, the input space $\mathcal{T} = \mathbb{R}$ is time, or $\mathcal{T} = \mathbb{R}^d$ is some feature space. An MOGP $f \sim \mathcal{GP}(m, K)$ is described by a *vector-valued* mean function

$$m(t) = \mathbb{E}[f(t)] = \begin{bmatrix}\mathbb{E}[f_1(t)] & \cdots & \mathbb{E}[f_p(t)]\end{bmatrix}^\mathsf{T}$$

and a *matrix-valued* covariance function

$$\begin{aligned} K(t,t') &= \mathbb{E}[f(t)f^\mathsf{T}(t')] - \mathbb{E}[f(t)]\mathbb{E}[f^\mathsf{T}(t')] \\ &= \begin{bmatrix} \operatorname{cov}(f_1(t), f_1(t')) & \cdots & \operatorname{cov}(f_1(t), f_p(t')) \\ \vdots & \ddots & \vdots \\ \operatorname{cov}(f_p(t), f_1(t')) & \cdots & \operatorname{cov}(f_p(t), f_p(t')) \end{bmatrix}. \end{aligned}$$

For n observations $y(t_1), \ldots, y(t_n) \in \mathbb{R}^p$, inference and learning take $O(n^3p^3)$ time and $O(n^2p^2)$ memory.

2.1 Mixing Model Hierarchy

Our unifying *Mixing Model Hierarchy* begins with Linear Mixing Models (LMMs), a simple class of MOGP models typically characterised by low-rank covariance structure. An LMM decomposes a signal $f(t)$ comprising p outputs into a fixed basis $h_1, \ldots, h_m \in \mathbb{R}^p$ whose coefficients $x_1(t), \ldots, x_m(t)$ are time-varying and modelled independently with Gaussian processes:

$$\begin{aligned} f(t) &= h_1x_1(t) + \ldots + h_mx_m(t) \\ &= \underbrace{\begin{bmatrix}h_1 & \cdots & h_m\end{bmatrix}}_{H} \underbrace{\begin{bmatrix}x_1(t) & \cdots & x_m(t)\end{bmatrix}^\mathsf{T}}_{x(t)} = Hx(t). \end{aligned}$$

The noisy signal $y(t)$ is, then, generated by adding $\mathcal{N}(0,\Lambda)$-distributed noise to $f(t)$. Intuitively, this means that the p-dimensional data lives in a "pancake" around the m-dimensional column space of H, where typically $m \ll p$.

Model 1 (Linear Mixing Model)**.** Let K be an $m \times m$ diagonal multi-output kernel with $K(t,t) = I_m$, H a $p \times m$ matrix, and $\Lambda = \operatorname{diag}(\sigma_1^2, \ldots, \sigma_p^2)$. Then the LMM is given by the following generative model:

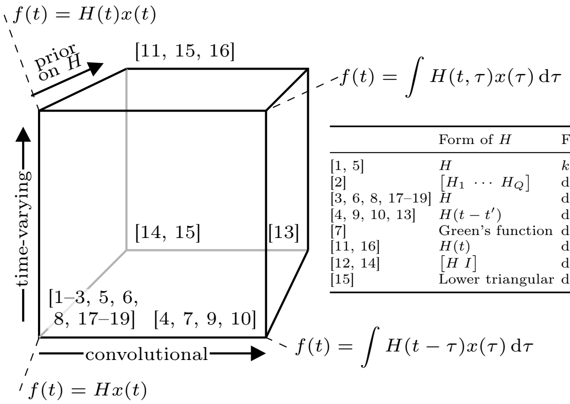
$$\begin{aligned} x &\sim \mathcal{GP}(0, K(t,t')), && \text{(latent processes)} \\ f(t) \,|\, H, x &= Hx(t), && \text{(likelihood)} \\ y \,|\, f &\sim \mathcal{GP}(f(t), \delta[t-t']\Lambda). && \text{(noise model)} \end{aligned}$$

We call x the *latent processes* and H the *mixing matrix* or *basis*. If we marginalise out f and x, we find $y \sim \mathcal{GP}(0, HK(t,t')H^\mathsf{T} + \Lambda)$, which reveals the low-rank covariance structure of the LMM. It also shows that the LMM is a time-varying generalisation of factor analysis (FA): choosing $K(t,t') = \delta[t-t']I_m$ recovers FA exactly.

The graphical model of the LMM is illustrated in the top-left corner of Fig. 6 in App. A, which highlights two restrictions of the LMM compared to a general MOGP: *(i)* the *instantaneous spatial covariance* of f, $\mathbb{E}[f(t)f^\mathsf{T}(t)] = HH^\mathsf{T}$, does not vary with time, because H does not vary with time; and *(ii)* the noise-free observation $f(t)$ is generated from $x(t')$ for $t' = t$ only, meaning that, for example, f cannot be x with a delay or a smoothed version of x. We hence call the LMM a *time-invariant* (due to *(i)*) and *instantaneous* (due to *(ii)*) MOGP.

The LMM can be generalised in three ways. First, the mixing matrix H may vary with time. Then $H \in \mathbb{R}^{p\times m}$ becomes a matrix-valued function $H\colon \mathcal{T} \to \mathbb{R}^{p\times m}$, and the likelihood becomes $f(t) \,|\, H, x = H(t)x(t)$. We call such MOGP models *time-varying* (see Fig. 6, top-right). Second, $f(t)$ may depend on $x(t')$ for all $t' \in \mathcal{T}$. Then the mixing matrix $H \in \mathbb{R}^{p\times m}$ becomes a matrix-valued time-invariant filter $H\colon \mathcal{T} \to \mathbb{R}^{p\times m}$, and the likelihood becomes $f(t) \,|\, H, x = \int H(t-\tau)x(\tau)\,\mathrm{d}\tau$. We call such MOGP models *convolutional* (see Fig. 6, bottom-left). Finally, $f(t)$ may depend on $x(t')$ for all $t' \in \mathcal{T}$ *and* this relationship may vary with time. Then the mixing matrix $H \in \mathbb{R}^{p\times m}$ becomes a matrix-valued time-varying filter $H\colon \mathcal{T}\times\mathcal{T} \to \mathbb{R}^{p\times m}$, and the likelihood becomes $f(t) \,|\, H, x = \int H(t,\tau)x(\tau)\,\mathrm{d}\tau$. We call such MOGP models *time-varying* and *convolutional*

[1] The distribution of $Ax + \varepsilon$ for A tall, x Gaussian, and ε noise (Roweis and Ghahramani, 1999; MacKay, 2002).



	Form of H	Form of K	Mixing
[1, 5]	H	kI	Inst.
[2]	$[H_1 \; \cdots \; H_Q]$	$\operatorname{diag}(k_1 I, \ldots, k_Q I)$	Inst.
[3, 6, 8, 17–19]	H	$\operatorname{diag}(k_1, \ldots, k_Q)$	Inst.
[4, 9, 10, 13]	$H(t-t')$	$\operatorname{diag}(\delta, \ldots, \delta)$	Conv.
[7]	Green's function	$\operatorname{diag}(k_1, \ldots, k_Q)$	Conv.
[11, 16]	$H(t)$	$\operatorname{diag}(k_1, \ldots, k_Q)$	Inst.
[12, 14]	$[H \; I]$	$\operatorname{diag}(k_1, \ldots, k_{Q+P})$	Inst.
[15]	Lower triangular	$\operatorname{diag}(k_1, \ldots, k_Q)$	Inst.

Models:

[1] Intrinstic Coregionalisation Model (Goovaerts, 1997)
[2] Linear Model of Coregionalisation (Goovaerts, 1997)
[3] Semiparametric Latent Factor Model (Teh and Seeger, 2005)
[4] Dependent Gaussian Processes (Boyle and Frean, 2005)
[5] Multi-Task Gaussian Processes (Bonilla et al., 2008)
[6] Osborne et al. (2008)
[7] Latent Force Models (Álvarez et al., 2009)
[8] Gaussian Process Factor Analysis (Yu et al., 2009)
[9] Efficient Multioutput Gaussian Processes Through Variational Inducing Kernels (Álvarez et al., 2010)
[10] Convolved Multiple Output Gaussian Processes (Álvarez and Lawrence, 2011)
[11] Gaussian Process Regression Network (Wilson et al., 2012)
[12] Collaborative Multi-Output Gaussian Processes (Nguyen and Bonilla, 2014)
[13] Generalised Gaussian Process Convolution Model (Bruinsma, 2016)
[14] Semi-Parametric Network Structure Discovery Models (Dezfouli et al., 2017)
[15] The Gaussian Process Autoregressive Regression Model (Requeima et al., 2019)
[16] Grouped Gaussian Processes (Dahl and Bonilla, 2019)
[17] Spatio-temporal Bayesian filtering and smoothing (Särkkä et al., 2013)
[18] Linear Mixing Model (Mod. 1)
[19] Orthogonal Linear Mixing Model (Mod. 2)

Figure 1: The Mixing Model Hierarchy, which organises multi-output GPs from the machine learning and geostatistics literature according to their modelling assumptions. In the table, $k_i = k_i(t,t')$ denotes a kernel, $\delta = \delta(t-t')$ is the Dirac delta function, "inst." abbreviates "instantaneous", and "conv." abbreviates "convolutional".

(see Fig. 6, bottom-right). The graphical models corresponding to these generalisations of the LMM are depicted in Fig. 6 in the App. A.

The LMM can be extended in one other way, which is to include a prior distribution on H. This extension and the three previously proposed generalisations together form the *Mixing Model Hierarchy* (MMH), which is depicted in Fig. 1. The MMH organises multi-output Gaussian process models according to their distinctive modelling assumptions. Fig. 1 shows how sixteen MOGP models from the machine learning and geostatistics literature can be recovered as special cases of the various generalisations of the LMM. Not all multi-output Gaussian process models are covered by the MMH, however. For example, Deep GPs (Damianou, 2015) and variations thereon (Kaiser et al., 2018) are excluded because they transform the latent processes *nonlinearly* to generate the observations.

2.2 Inference and Learning in the LMM

In MOGPs, the complexities of inference and learning can often be alleviated by exploiting structure in the kernel. This is also the case for the Linear Mixing Model: the p-dimensional observations have only m degrees of freedom, where typically $m \ll p$, and it turns out that the $O(n^3p^3)$ time and $O(n^2p^2)$ memory complexities can be brought down to $O(n^3m^3)$ and $O(n^2m^2)$. In particular, in the following we show that the covariance structure of the LMM allows us to summarise the p-dimensional observations in an m-dimensional space, without loss of information.

We identify a low-dimensional projection of observations y that is a sufficient statistic for the model and which can therefore be used to accelerate inference. This projection of y is given by the maximum likelihood estimate (MLE) of x under the likelihood of the LMM. Denote $p(y \,|\, x) = \mathcal{N}(y; Hx, \Lambda)$, and let T be the $m \times p$ matrix $(H^\top \Lambda^{-1} H)^{-1} H^\top \Lambda^{-1}$. Prop. 3 in App. B shows that $Ty = \arg\max_x \, p(y \,|\, x)$ and that Ty is an unbiased estimate of x. Hence, Ty is a MLE for x, which means that it is a function of a sufficient statistic for x, if one exists. The following proposition shows that Ty is actually minimally sufficient itself.

Prop. 1. The MLE Ty of x is a minimal sufficient statistic for x. *Proof.* See App. C.

For any prior $p(x)$ over x, sufficiency of T gives that $p(x \,|\, y) = p(x \,|\, Ty)$; *i.e.*, conditioning on y is equivalent to conditioning on Ty, where Ty can be interpreted as a "summary" or "projection" of y. This idea is formalised in the following proposition, which is an immediate consequence of Prop. 1 and Lem. 1 in App. C.

Prop. 2. Let $p(x)$ be a model for $x\colon \mathcal{T} \to \mathbb{R}^m$, not necessarily Gaussian, H a $p \times m$ matrix, and $\Lambda = \operatorname{diag}(\sigma_1^2, \ldots, \sigma_p^2)$. Then consider the following generative model:

$$\begin{aligned} x &\sim p(x), && \text{(latent processes)} \\ f(t) \,|\, H, x &= Hx(t), && \text{(likelihood)} \\ y \,|\, f &\sim \mathcal{GP}(f(t), \delta[t-t']\Lambda). && \text{(noise model)} \end{aligned}$$

Consider a $p \times n$ matrix Y of observations of y. Then $p(f \,|\, Y) = p(f \,|\, TY)$. Furthermore, the distribution of the projected observed signal Ty is given by

$$Ty \,|\, x \sim \mathcal{GP}(x(t), \delta[t-t']\Lambda_T) \text{ with } \Lambda_T = (H^\top \Lambda^{-1} H)^{-1},$$

and the likelihood of $Y = (y_i)_{i=1}^n \subseteq \mathbb{R}^p$ is given by

$$p(Y) = \left[\prod_{i=1}^n \frac{\mathcal{N}(y_i; 0, \Lambda)}{\mathcal{N}(y_i; 0, \Lambda_T)}\right] \int p(x) \prod_{i=1}^n \mathcal{N}(Ty_i; x_i, \Lambda_T)\,\mathrm{d}x.$$

Crucially, Y are p-dimensional observations, TY are m-dimensional summaries, and typically $m \ll p$, so conditioning on TY is often much cheaper. In particular, if we apply Prop. 2 to the LMM by letting $x \sim \mathcal{GP}(0, K(t,t'))$, we immediately get the claimed reduction in complexities: whereas conditioning on Y takes $O(n^3p^3)$ time and $O(n^2p^2)$ memory, we may equivalently condition on TY, which takes $O(n^3m^3)$ time and $O(n^2m^2)$ memory instead. This important observation is depicted in Fig. 2a.

In Prop. 2, we call $\Lambda_T = T\Lambda T^\top = (H^\top \Lambda^{-1} H)^{-1}$ the *projected observation noise*. The noise term Λ_T is important, because it couples the latent processes upon observing data. In particular, if the latent processes are independent under the prior and Λ_T is diagonal, then the latent processes remain independent when data is observed. This observation forms the basis of the computational gains achieved by the Orthogonal Linear Mixing Model, which we present next.

3 THE ORTHOGONAL LINEAR MIXING MODEL

The Orthogonal Linear Mixing Model (OLMM) is the special case of the LMM where the basis, the columns of H, is *orthogonal*. In particular, $H = US^{\frac{1}{2}}$ where U is a truncated orthogonal matrix and $S > 0$ a diagonal scaling. We define this model as follows.

Model 2 (Orthogonal Linear Mixing Model)**.** Let K be an $m \times m$ diagonal multi-output kernel with $K(t,t) = I_m$, H a $p \times m$ matrix of the form $H = US^{\frac{1}{2}}$ with U a truncated orthogonal matrix and $S > 0$ diagonal, and $\Lambda = \sigma^2 I_p + HDH^\top$ a $p \times p$ matrix with $D > 0$ diagonal. Then the OLMM is given by the following generative model:

$$\begin{aligned} x &\sim \mathcal{GP}(0, K(t,t')), && \text{(latent processes)} \\ f(t) \,|\, H, x &= Hx(t), && \text{(likelihood)} \\ y \,|\, f &\sim \mathcal{GP}(f(t), \delta[t-t']\Lambda). && \text{(noise model)} \end{aligned}$$

The difference between the LMM and the OLMM is illustrated in Fig. 2c. In the OLMM, the number of latent processes must be smaller than or equal to the number of outputs, since the number of p-dimensional vectors that can be mutually orthogonal is at most p. Also, D in Λ can be interpreted as heterogeneous noise deriving from the latent processes. Moreover, although H and Λ do not depend on time, our analysis and results also apply to the case where H_t and Λ_t do vary with time. Finally, for the OLMM, Prop. 5 in App. E shows that $T = S^{-\frac{1}{2}}U^\top$ and $\Lambda_T = \sigma^2 S^{-1} + D$.

Whereas the LMM is a time-varying generalisation of FA, the OLMM is a time-varying generalisation of probabilistic principal component analysis (PPCA, Tipping and Bishop, 1999): $D = 0$ and $K(t,t') = \delta[t-t']I_m$ recovers PPCA exactly. The OLMM is also similar to Gaussian Process Factor Analysis (GPFA, Yu et al., 2009), with the crucial difference being that in GPFA orthogonalisation of the columns of H is done as a post-processing step, whereas in the OLMM orthogonality of the columns of H is built into the model.

Generality of the OLMM. In the separable case, where $K(t,t') = k(t,t')I_m$ for a scalar-valued kernel k, for every LMM with homogeneous observation noise ($\Lambda = \sigma^2 I_p$), there exists an OLMM with $D = 0$ that is equal in distribution of y. To see this, note that

$$\begin{aligned} y^{(\text{LMM})} &\sim \mathcal{GP}(0, k(t,t')HH^\top + \sigma^2\delta[t-t']I_p), \\ y^{(\text{OLMM})} &\sim \mathcal{GP}(0, k(t,t')USU^\top + \sigma^2\delta[t-t']I_p). \end{aligned}$$

Hence, letting $USU^\top$ be the eigendecomposition of $HH^\top$ gives an OLMM equal in distribution of y. In the nonseparable case, where diagonal elements of K are linearly independent, in general only the distribution of $y(t)$ at every t can be recovered by an OLMM, but the correlation between $y(t)$ and $y(t')$ for $t' \neq t$ may be different. In terms of the joint distribution over x and y, which is important for interpretability of the latent processes, App. G shows that the Kullback–Leibler (KL) divergence between two LMMs with bases H and $\hat{H}$ is proportional to $\|H - \hat{H}\|_F^2$, hence symmetric, where $\|\bullet\|_F$ denotes the Frobenius norm. As a consequence (App. G), the KL between an LMM with basis H and the OLMM closest in KL is upper bounded by $\|I - V\|_F^2$ where V are the right singular vectors of H. This makes sense: $V = I$ implies that H is of the form $US^{\frac{1}{2}}$ with U a truncated orthogonal matrix and $S > 0$ diagonal. It also shows that an LMM is close to an OLMM if V is close to I in the sense of the Frobenius norm.

Choice of H. The basis H can be initialised and learned in several ways. *(i)* H can be initialised randomly and learned through gradient-based optimisation of the marginal likelihood. *(ii)* Using that $\mathbb{E}[f(t)f^\top(t)] = HH^\top$, the basis H can be initialised to $\hat{U}\hat{S}^{\frac{1}{2}}$ where $\hat{\Sigma} = \hat{U}\hat{S}\hat{U}^\top$ is the eigendecomposition of an estimate $\hat{\Sigma}$ of the spatial covariance. *(iii)* In the case that there is a kernel over the outputs, e.g. in separable spatio-temporal GP models, H can be set to $US^{\frac{1}{2}}$ where $USU^\top$ is an eigendecomposition of the kernel matrix over the outputs. The hyperparameters of the kernel over the outputs can then be learned with gradient-based optimisation by differentiating through the eigendecomposition. We discuss this choice in more detail in Sec. 4.

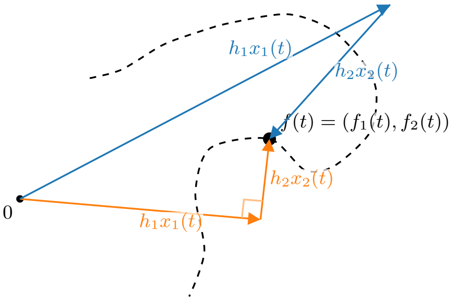
Diagonal projected noise. As alluded to in Sec. 2.2, the most important property of the OLMM is that the projected noise Λ_T from Prop. 2 is diagonal: $\Lambda_T =$

$$\begin{tikzcd}
Y \arrow[r, "\text{inference}", "O(n^3p^3)"'] \arrow[d] & p(f \,|\, Y) \\
TY \arrow[r, "\text{inference}", "O(n^3m^3)"'] & p(f \,|\, TY) \arrow[u]
\end{tikzcd}$$

(a) Conditioning on Y in the LMM

$$\begin{tikzcd}
Y \arrow[r, "\text{inference}", "O(n^3p^3)"'] \arrow[dd] & p(f \,|\, Y) \\
 & p(x_1 \,|\, (TY)_{1:}) \arrow[u, bend right] \\
TY \arrow[ur, "\text{inference}", "O(n^3)"'] \arrow[dr, "\text{inference}", "O(n^3)"'] & \vdots \\
 & p(x_m \,|\, (TY)_{m:}) \arrow[uuu, bend right]
\end{tikzcd}$$

(b) Conditioning on Y in the OLMM



(c) Difference between LMM and OLMM

Figure 2: (a–b) Commutative diagram depicting that conditioning on Y in the LMM and in the OLMM is equivalent to independently conditioning every x_i on $(TY)_{i:}$, but yields different computational complexities. (c) Illustration of the difference between the LMM and OLMM. The trajectory of a particle in two dimensions is modelled by the LMM (blue) and OLMM (orange). The noise-free position $f(t)$ is modelled as a linear combination of basis vectors h_1 and h_2 with coefficients $x_1(t)$ and $x_2(t)$ (two independent GPs). In the OLMM, the basis vectors h_1 and h_2 are constrained to be orthogonal, whereas in the LMM, h_1 and h_2 are unconstrained.

$\sigma^2 S^{-1} + D$. Prop. 4 in the App. D characterises exactly when this is the case. Specifically, Prop. 4, says that the projected noise is diagonal if and only if H is of the form $H = \Lambda^{\frac{1}{2}} U S^{\frac{1}{2}}$ with U a truncated orthogonal matrix and $S > 0$ diagonal. This condition is awkward, as it couples H and Λ. Fortunately, Prop. 4 also shows that we may drop H's dependency on Λ if and only if every column of U is an eigenvector of Λ. In the OLMM, every column of U is indeed an eigenvector of Λ. We therefore identify the OLMM as a general parametrisation of the LMM for which the projected noise is diagonal. Compared to the LMM, this brings computational advantages, which we discuss next.

Inference. Since the projected noise is diagonal, the latent processes remain independent when data is observed. We may hence treat the latent processes independently, conditioning the i^{th} latent process x_i on $(TY)_{i:} = Y^\top u_i / \sqrt{S_{ii}}$ under noise $(\Lambda_T)_{ii} = \sigma^2/S_{ii} + D_{ii}$, where u_i is the i^{th} column of U. Therefore, inference takes $O(n^3 m)$ time and $O(n^2 m)$ memory, which are *linear* in the number of degrees of freedom in the data m. This decoupled inference procedure is depicted in Fig. 2b. Remark that the decoupled problems can be treated in parallel to achieve sublinear wall time, and that in the separable case further speedups are possible.

Learning. For computing the likelihood of observed data, the OLMM also offers computational benefits. As the projected noise is diagonal, by absorbing D into K, the expression for $p(Y)$ from Prop. 1 further simplifies to the following:

$$\begin{aligned} p(Y) &= \mathcal{N}(\operatorname{vec} Y; 0, \sigma^2 I_{np}) \\ &\quad \times \prod_{i=1}^{m} \frac{\mathcal{N}((TY)_{i:}; 0, K_i + (\sigma^2/S_{ii} + D_{ii}) I_n)}{\mathcal{N}((TY)_{i:}; 0, (\sigma^2/S_{ii}) I_n)}, \end{aligned}$$

where K_i is the $n \times n$ kernel matrix for the i^{th} latent process x_i. We conclude that learning also takes $O(n^3 m)$ time and $O(n^2 m)$ memory, again *linear* in the number of latent processes. Furthermore, we observe that $p(Y)$ is *convex* in U, which significantly aids learning; in comparison, for the LMM, $p(Y)$ is highly nonconvex in the mixing matrix H.

Interpretability. Besides computational benefits, that the OLMM breaks down into independent problems for the latent processes also promotes interpretability. Namely, the independent problems can be separately inspected to interpret, diagnose, and improve the model. This is *much* easier than directly working with predictions for the data, which are high dimensional and often strongly correlated between outputs. In particular, if H is sufficiently rich and the important independent problems are solved sufficiently well, then the OLMM achieves good mean squared error (MSE) for the data:

$$\underbrace{\|y - Hx\|^2}_{\text{MSE}} = \underbrace{\|P_{U^\perp} y\|^2}_{\substack{\text{variance not} \\ \text{captured by basis}}} + \sum_{i=1}^{m} S_{ii} \underbrace{((Ty)_i - x_i)^2}_{\substack{\text{MSE of} \\ i^{\text{th}} \text{ latent process}}}$$

where $P_{U^\perp}$ is the orthogonal projection onto the orthogonal complement of U. See App. F for a proof.

Scaling. For both learning and inference, the problem decouples into m independent single-output problems. Therefore, to scale to a large number of data points n, off-the-shelf single-output GP scaling techniques can be trivially applied to these independent problems. For example, if the variational inducing point method by Titsias (2009) is used with $r \ll n$ inducing points, then inference and learning are further reduced to $O(nr^2 m)$ time and $O(nrm)$ memory. Most importantly, if $k(t, t')$ is Markovian (e.g. of the Matérn class) one can leverage state-space methods to exactly solve the m independent problems in closed-form (Hartikainen and Särkkä, 2010;

Särkkä and Solin, 2019). This brings down the scaling to $O(nmd^3)$ time and $O(nmd^2)$ memory, where d is the state dimension (typically $d \ll m, n$). We further discuss this approach in Sec. 4.

Missing data and heterogeneous observation noise. Missing data is troublesome for the OLMM, because it is not possible to take away a subset of the rows of H and retain orthogonality of the columns. Possible methods to circumvent this issue are discussed in App. H, but are not explored further in this work. Furthermore, the specification of the OLMM does not allow for heterogeneous observation noise. This issue is discussed in App. I, which shows that the OLMM can have automatic whitening of the data built in to allow for heterogeneous observation noise.

4 APPLICATION TO SEPARABLE SPATIO-TEMPORAL MODELS

Separable spatio-temporal GPs, which are of the form $f \sim \mathcal{GP}(0, k_t(t,t')k_r(r,r'))$, fall neatly within the OLMM framework if observed at a fixed number of locations in space. Specifically, the vector-valued process $f(t) = (f(t, r_i))_{r=1}^{p}$ has distribution $\mathcal{GP}(0, k_t(t,t')K_r)$ where K_r is the $p \times p$ matrix with $(K_r)_{ij} = k_r(r_i, r_j)$. Letting USU^{T} be the eigendecomposition of K_r, $f(t)$ is an OLMM with $H = US^{\frac{1}{2}}$, which we note is square, and $K(t,t') = k_t(t,t')I_p$. Note that, in this case, projecting the data takes $O(np^2)$ time. By relaxing K to be a general diagonal multi-output kernel with $K(t,t) = I_p$, we obtain a nonseparable relaxation of the original separable model whilst retaining exact, computationally efficient inference. In particular, the orthogonal basis for the OLMM are the eigenvectors of a kernel matrix whose hyperparameters can be optimised.

Combining the OLMM framework with efficient state-space scaling techniques (Hartikainen and Särkkä, 2010; Särkkä and Solin, 2019; Solin et al., 2018; Nickisch et al., 2018), which are either exact or arbitrarily good approximations, the complexities are reduced to $O(np+p^3)$ time and $O(np+p^2)$ memory for the entire problem, which are *linear* in n. This compares favourably with the filtering approach by Särkkä et al. (2013), which takes $O(np^3)$ time and $O(np^2)$ memory. The sparse approximation by Hartikainen et al. (2011) provides a reduced-rank approximation with $O(nq^3)$ time and $O(nq^2)$ memory for some $q \ll p$, although it comes at the expense of accuracy and yields no asymptotic savings if q must increase linearly in p to maintain a reasonable accuracy. It also compares favourably with the Kronecker product decomposition (Saatçi, 2012, Ch. 5) approach, which requires $O(p^3 + n^3)$ time and $O(p^2 + n^2)$ memory complexity.

5 EXPERIMENTS

We test the OLMM in experiments on synthetic and real-world data sets. An implementation is available at `https://github.com/wesselb/olmm`.

Computational scaling. We demonstrate that exact inference scales favourably in m for the OLMM, whereas the LMM quickly becomes computationally infeasible as m increases. We use a highly optimised implementation of exact inference for the LMM, kindly made available by Invenia Labs[2]. Fig. 3 shows the runtime and the memory usage of the LMM and OLMM. Observe that the LMM scales $O(m^3)$ in time and $O(m^2)$ in memory, whereas the OLMM scales $O(m)$ in both time and memory. For $m = 25$, the LMM takes nearly 10 minutes to compute the evidence, whereas the OLMM only requires a couple seconds.[3]

Temperature extrapolation. We consider a simple spatio-temporal temperature prediction problem over Europe. Approximately 30 years worth of the ERA-Interim reanalysis temperature data[4] (Dee et al., 2011) is smoothed in time with a Hamming window of width 31 and sub-sampled once every 31 days to produce a data set comprising $13 \times 19 = 247$ outputs and approximately 350 months worth of data. We train the OLMM and independent GPs (both models use Matérn–5/2 kernels with a periodic component) on the first 250 months of the data and test on the next 100 months. The basis for the OLMM is given by the eigenvectors of the kernel matrix over the points in space (Matérn–5/2 with a different length scale for latitude and longitude). Table 1 shows the log-marginal likelihood (LML) of the training data, the posterior predictive log-probability (PPLP) and the root-mean-square error (RMSE) of the test data. Although the RMSE is equal to that of the independent GPs—the data is highly periodic and the predictions are highly accurate for both models—the LML and PPLP are much better, which indicates that that the OLMM is able to capture meaningful structure between temperatures at different points in space.

Rainforest tree point process modelling. We consider a subset of the extensive rainforest data set credited to Hubbell et al. (2005); Condit (1998); Hubbell

[2] `https://invenialabs.co.uk`

[3] Measurements were performed using a MacBook Pro with a 2.7 GHz Intel Core i7 processor and 16 GB RAM. Code was implemented in Julia 1.0 (Bezanson et al., 2017) and memory and time were measured using the `@allocated` and the `@elapsed` macros, respectively, with the measurements averaged over 10 samples run serially. This means memory reported is the total memory allocated, not peak memory consumption.

[4] All output from CMIP5 and ERA-Interim models was regridded onto the latitude–longitude grid used for the IPSL-CM5A-LR model.

	LML	PPLP	RMSE
IGP	−0.247	−0.219	1.99
OLMM	1.22	1.19	1.99

Table 1: Log-marginal likelihood (LML) of the training data, posterior predictive log-probability (PPLP) and root-mean-square error (RMSE) of held-out test data of independent GPs (IGP) and the OLMM for the temperature extrapolation experiments.

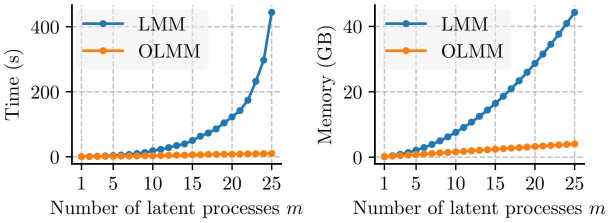


Figure 3: Runtime (left) and memory usage (right) of the LMM and OLMM for computing the evidence of $n = 1500$ observations for $p = 200$ outputs

et al. (1999). The data features a 1000 m × 500 m rainforest dynamics plot in Barro Colorado Island, Panama. In the survey area, the locations of all *Trichilia tuberculata* (a tree species of the Mahogany family) have been measured (see Fig. 7 in App. J). We tackle this spatial point pattern with a log-Gaussian Cox process model, which is an inhomogeneous Poisson process model for count data. The unknown intensity function $\lambda(x)$ is modelled with a Gaussian process such that $f(x) = \log \lambda(x)$. We model locally constant intensity in subregions by discretising the region into np bins (Møller et al., 1998). This leads to having a Poisson observation model in each bin (see App. J). This model reaches posterior consistency in the limit of bin width going to zero (Tokdar and Ghosh, 2007). The accuracy thus improves with tighter binning. We use a separable Matérn-5/2 GP prior over $f(x_1, x_2)$, and discretise the area into a $n \times p = 200 \times 100$ grid with $np = 20000$ grid bins, and treat the first dimension as time.

We use Elliptical Slice Sampling (Murray et al., 2010; Murray and Adams, 2010) and Metropolis–Hastings (Hastings, 1970) to perform inference in this problem, which require many evaluations of the log marginal likelihood and draws from the prior. The proposed methodology renders these operations much less computationally intensive than the general case. We find approximately an hour is required to perform 10^5 MCMC iterations on a MacBook Pro[5]. See App. J for more details.

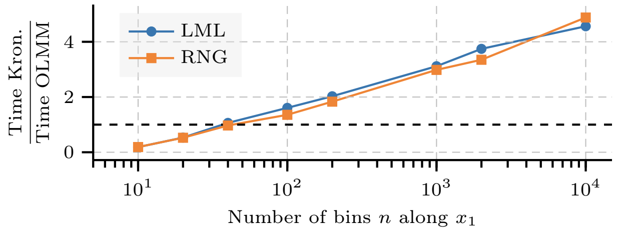
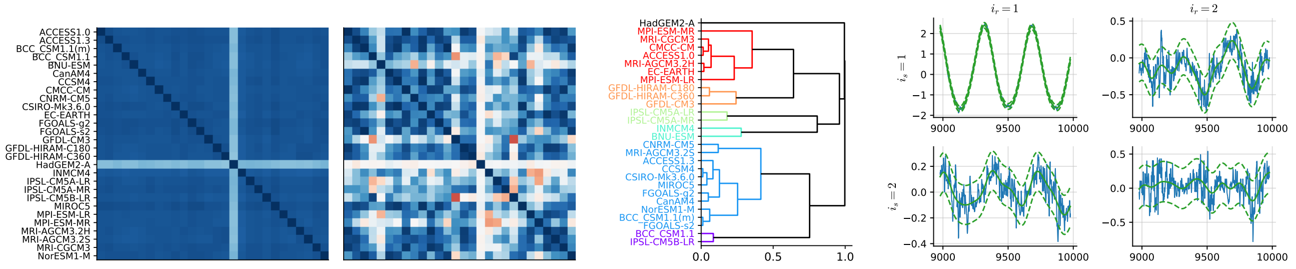


Figure 4: Ratio of timings of the Kronecker approach (Saatçi, 2012, Ch. 5) and the OLMM to compute the log marginal likelihood (LML) and generate a single prior sample (RNG)

As previously discussed, the Kronecker product factorisation techniques (Saatçi, 2012, Ch. 5) are applicable in separable spatio-temporal scenarios. While our method in combination with state-space inference offers superior performance asymptotically, we demonstrate that we also obtain benefits on practical problems. Fig. 4 shows for different number of bins n (with $p = n/2$) the time for log marginal likelihood evaluation and sampling from the GP prior for our method and for the Kronecker product decomposition approach. Observe that the linear scaling of our methodology quickly renders it competitive. While this simple experiment does not provide a comprehensive assessment of the relative merits of these techniques—for example, only a single CPU core was used in this experiment—it clearly establishes the practical relevance of our technique in the separable setting.

Large-scale climate model calibration. In this final experiment, we consider a large-scale climate modelling task over Europe. In particular, we use the OLMM to find relationships between 28 climate simulators[4] (see Taylor et al., 2012, for background) by letting $H = H_s \otimes H_r$, where H_s are the first $m_s = 5$ eigenvectors of a 28×28 covariance matrix K_s between the simulators, and H_r are the first $m_r = 10$ eigenvectors of the kernel matrix over the points in space (Matérn–5/2 with a different length scale for latitude and longitude). This means the $m_r m_s = 50$ latent processes are indexed by two indices i_s and i_m, one corresponding to the eigenvector of the simulator covariance and one correponding to the eigenvector of the spatial covariance. In the above, the covariance matrix K_s between the simulators is a parameter that we learn. The kernels for the latent processes are Matérn–5/2. We consider $n = 10000$ time points for all 28 simulators, each with 247 outputs, giving a total of roughly 70 million data points. For the independent problems, we use the variational inducing point method by Titsias (2009), where the positions of the inducing points are initialised to one every two months. All hyperparameters and the locations of the inducing points are optimised until convergence using `scipy`'s implementation of the L-BFGS-B algorithm (Nocedal and Wright,

[5] 2.5 GHz Intel Core i7 processor and 16 GB RAM



(a) Empirical and learned correlations

(b) Clustering

(c) Latent processes

Figure 5: Results of the large-scale climate simulator experiment, showing (a) the empirical and learned correlation between the simulators, (b) a dendrogram deriving from hierarhically clustering the simulators based on the learned correlations, and (c) predictions for the latent processes for the first two eigenvectors of the covariance matrix between simulators $i_s = 1, 2$ and the first two eigenvectors of the spatial covariance $i_r = 1, 2$, for the last 1000 days.

2006), which takes about 4 hours on a MacBook Pro[5]. The learned length scales were 23.3° for latitude and 43.6° for longitude.

Fig. 5a shows the empirical correlations and the correlations learned by the OLMM (derived from K_s). Observe that the empirical correlation is not informative, as it ignores all temporal structure. The OLMM, however, does not, and is consequently able to identify a rich correlation structure. In order to get insight into the learned correlations, we hierarchically cluster the models using farthest point linkage with $1 - |\text{corr.}|$ as the distance. Fig. 5b shows the resulting dendrogram, in which models are grouped by their similarity. For two models, the further to the right the branch connecting them is, the less similar the models are. In Figs. 5a and 5b, HadGEM2 is clearly singled out: it is one of the simplest models, not including several processes that can be found in others, such as ocean & sea-ice, terrestrial carbon cycle, stratosphere, and ocean biogeochemistry (Bellouin et al., 2011). Furthermore, if we inspect the names of the simulators in the groups in Fig. 5b, we observe that often simulators of the same family are grouped together. We observe some interesting cases. *(i)* Although IPSL-CM5A-LR and IPSL-CM5A-MR are close, IPSL-CM5B-LR is grouped far apart. It turns out that IPSL-CM5A-LR and IPSL-CM5A-MR are different-resolution versions of the same model, while IPSL-CM5B-LR employs a different atmospheric model.[6] *(ii)* ACCESS1.0 and ACCESS1.3 have a similar name, but differ greatly in their implementation: ACCESS1.0 is the basic model, while ACCESS1.3 is much more aspirational, including experimental atmospheric physics models and a particular land surface model (Bi et al., 2013). *(iii)* The distance between BCC_CSM1.1(m) and BCC_CSM1.1 can be explained by the more realistic surface air temperature predictions obtained by the former (Wu et al., 2014), which is exactly the quantity we study. Finally, Fig. 5c shows predictions for four latent processes ($i_s = 1, 2$ with $i_r = 1, 2$). The first spatial eigenvector ($i_r = 1$) is constant in space; combined with the strongest eigenvector of K_s ($i_s = 1$), we obtain a strong signal constituting seasonal temperature changes.

6 DISCUSSION AND CONCLUSION

In this paper, we introduced the Orthogonal Linear Mixing Model as a multi-output GP in the Linear Mixing Model class where exact inference and learning scales linearly in the desired number m of degrees of freedom of the observations. The OLMM is simple to implement and trivially compatible with off-the-shelf single-output GP scaling techniques to scale to large numbers of observations n. We introduced the Mixing Model Hierarchy, which organises MOGPs according to their generative models, and placed the OLMM within this scheme.

The OLMM is exciting for several reasons. It provides a way to do efficient inference in separable spatio-temporal models that is trivially compatible with single-output scaling techniques. Exact inference is *linear* in number of outputs, and, with state space methods, it is also *linear* in number of time points. The OLMM even extends separable spatio-temporal models to non-separable models whilst retaining exact, computationally efficient inference. Furthermore, the OLMM is interpretable: the latent processes are independent, which means that they can be inspected separately to diagnose the model. Finally, as demonstrated in the real-world experiments in the paper, the OLMM is applicable to a wide range of modelling tasks. An interesting future direction for the method would be targeting sub-linear time complexity by parallelisation (see, e.g., Särkkä and García-Fernández, 2019).

[6] `https://portal.enes.org/models/earthsystem-models/ipsl/ipslesm`

Bibliography

M. Álvarez and N. D. Lawrence. Sparse convolved Gaussian processes for multi-output regression. volume 21, pages 57–64. Curran Associates, Inc., 2009.

M. Álvarez, D. Luengo, and N. Lawrence. Latent force models. 5:9–16, 2009.

M. Álvarez, D. Luengo, M. Titsias, and N. D. Lawrence. Efficient multioutput Gaussian processes through variational inducing kernels. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 25–32. PMLR, 2010.

M. A. Álvarez and N. D. Lawrence. Computationally efficient convolved multiple output Gaussian processes. *Journal of Machine Learning Research*, 12: 1459–1500, 7 2011.

N. Bellouin, W. Collins, I. Culverwell, P. Halloran, S. Hardiman, T. Hinton, C. Jones, R. McDonald, A. McLaren, F. O'Connor, et al. The hadgem2 family of met office unified model climate configurations. *Geoscientific Model Development*, 4(3): 723–757, 2011.

J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah. Julia: A fresh approach to numerical computing. *SIAM review*, 59(1):65–98, 2017.

D. Bi, M. Dix, S. J. Marsland, S. O'Farrell, H. Rashid, P. Uotila, A. Hirst, E. Kowalczyk, M. Golebiewski, A. Sullivan, et al. The access coupled model: description, control climate and evaluation. *Aust. Meteorol. Oceanogr. J*, 63(1):41–64, 2013.

E. V. Bonilla, F. V. Agakov, and C. K. I. Williams. Kernel multi-task learning using task-specific features. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, volume 2 of *Proceedings of Machine Learning Research*, pages 43–50. PMLR, 2007.

E. V. Bonilla, K. M. Chai, and C. K. I. Williams. Multi-task Gaussian process prediction. In *Advances in Neural Information Processing Systems*, volume 20, pages 153–160. MIT Press, 2008.

P. Boyle and M. Frean. Dependent Gaussian processes. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 217–224. MIT Press, 2005.

E. Brochu, V. M. Cora, and N. de Freitas. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 12 2010.

W. P. Bruinsma. The generalised Gaussian convolution process model. Master's thesis, Department of Engineering, University of Cambridge, 2016.

T. Bui, D. Hernández-Lobato, J. Hernandez-Lobato, Y. Li, and R. Turner. Deep Gaussian processes for regression using approximate expectation propagation. pages 1472–1481, 2016.

T. D. Bui, J. Yan, and R. E. Turner. A unifying framework for Gaussian process pseudo-point approximations using power expectation propagation. *Journal of Machine Learning Research*, 18(104):1–72, 2017.

J. Q. Candela and C. E. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6:1939–1959, 12 2005.

G. Casella and R. Berger. *Statistical Inference*. Duxbury Resource Center, 6 2001.

J. Chen and J. Revels. Robust benchmarking in noisy environments. *arXiv e-prints*, art. arXiv:1608.04295, Aug 2016.

C.-A. Cheng and B. Boots. Variational inference for Gaussian process models with linear complexity. In *Advances in Neural Information Processing Systems*, pages 5184–5194. Curran Associates, Inc., 2017.

R. Condit. *Tropical Forest Census Plots*. Springer-Verlag and R. G. Landes Company, Berlin, Germany, and Georgetown, Texas, 1998.

A. Dahl and E. V. Bonilla. Grouped Gaussian processes for solar power prediction. *Machine Learning*, 108 (8-9):1287–1306, 2019.

A. Damianou. *Deep Gaussian Processes and Variational Propagation of Uncertainty*. PhD thesis, Department of Neuroscience, University of Sheffield, 2015.

D. P. Dee, S. Uppala, A. Simmons, P. Berrisford, P. Poli, S. Kobayashi, U. Andrae, M. Balmaseda, G. Balsamo, d. P. Bauer, et al. The era-interim reanalysis: Configuration and performance of the data assimilation system. *Quarterly Journal of the royal meteorological society*, 137(656):553–597, 2011.

M. P. Deisenroth and C. E. Rasmussen. PILCO: A model-based and data-efficient approach to policy search. volume 28, pages 465–472. Omnipress, 2011.

A. Dezfouli, E. V. Bonilla, and R. Nock. Semi-parametric network structure discovery models. *arXiv preprint arXiv:1702.08530*, 2 2017.

D. Duvenaud. *Automatic Model Construction With Gaussian Processes*. PhD thesis, Computational and Biological Learning Laboratory, University of Cambridge, 2014.

P. Goovaerts. *Geostatistics for Natural Resources Evaluation.* Oxford University Press, 1 edition, 1997.

J. Hartikainen and S. Särkkä. Kalman filtering and smoothing solutions to temporal Gaussian process regression models. In *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 379–384, 2010.

J. Hartikainen, J. Riihimäki, and S. Särkkä. Sparse spatio-temporal gaussian processes with general likelihoods. In *International Conference on Artificial Neural Networks*, pages 193–200. Springer, 2011.

W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. 1970.

P. Hennig, M. A. Osborne, and M. Girolami. Probabilistic numerics and uncertainty in computations. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 471(2179), 2015.

J. Hensman, N. Fusi, and N. D. Lawrence. Gaussian processes for big data. In *Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence*, pages 282–290, 2013.

J. Hensman, N. Durrande, and A. Solin. Variational fourier features for Gaussian processes. *Journal of Machine Learning Research*, 18(151):1–52, 2018.

S. Hubbell, R. Foster, S. O'Brien, K. Harms, R. Condit, B. Wechsler, S. Wright, and S. De Lao. Light-gap disturbances, recruitment limitation, and tree diversity in a neotropical forest. *Science*, 283(5401):554–557, 1999.

S. Hubbell, R. Condit, and R. Foster. Barro colorado forest census plot data. URL: `https://ctfs.arnarb.harvard.edu/webatlas/datasets/bci`, 2005.

M. Kaiser, C. Otte, T. Runkler, and C. H. Ek. Bayesian alignments of warped multi-output Gaussian processes. In *Advances in Neural Information Processing Systems*, pages 6995–7004, 2018.

D. P. Kingma and M. Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 12 2013.

M. Lázaro-Gredilla, J. Q. Candela, C. E. Rasmussen, and A. R. Figueiras-Vidal. Sparse spectrum Gaussian process regression. *Journal of Machine Learning Research*, 11:1865–1881, 2010.

D. J. C. MacKay. *Information Theory, Inference & Learning Algorithms.* Cambridge University Press, 2002.

G. Matheron. Le krigeage universel. In *Cahiers du Centre de morphologie mathématique de Fontainebleau*, volume 1. École nationale supérieure des mines de Paris, 1969.

T. Minka. Deriving quadrature rules from Gaussian processes. Technical report, 2000.

J. Møller, A. R. Syversveen, and R. P. Waagepetersen. Log Gaussian Cox processes. *Scandinavian Journal of Statistics*, 25(3):451–482, 1998.

I. Murray and R. P. Adams. Slice sampling covariance hyperparameters of latent Gaussian models. In *Advances in Neural Information Processing Systems*, volume 23, pages 1732–1740. Curran Associates, Inc., 2010.

I. Murray, R. Adams, and D. MacKay. Elliptical slice sampling. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 541–548. PMLR, 13–15 May 2010.

T. V. Nguyen and E. V. Bonilla. Collaborative multi-output Gaussian processes. In *Conference on Uncertainty in Artificial Intelligence*, volume 30, 2014.

H. Nickisch, A. Solin, and A. Grigorievskiy. State space gaussian processes with non-gaussian likelihood. *arXiv preprint arXiv:1802.04846*, 2018.

J. Nocedal and S. J. Wright. *Numerical Optimization.* Springer, 2 edition, 2006.

M. A. Osborne, S. J. Roberts, A. Rogers, S. D. Ramchurn, and N. R. Jennings. Towards real-time information processing of sensor network data using computationally efficient multi-output Gaussian processes. In *Proceedings of the 7th International Conference on Information Processing in Sensor Networks*, IPSN '08, pages 109–120. IEEE Computer Society, 2008.

G. Parra and F. Tobar. Spectral mixture kernels for multi-output Gaussian processes. In *Advances in Neural Information Processing Systems*, pages 6681–6690. Curran Associates, Inc., 2017.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning.* MIT Press, 2006.

J. Requeima, W. Tebbutt, W. Bruinsma, and R. E. Turner. The Gaussian process autoregressive regression model (GPAR). In K. Chaudhuri and M. Sugiyama, editors, *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*, pages 1860–1869. PMLR, 4 2019.

S. Roweis and Z. Ghahramani. A unifying review of linear Gaussian models. *Neural Computation*, 11(2):305–345, 1999. doi: 10.1162/089976699300016674. URL `https://doi.org/10.1162/089976699300016674`.

Y. Saatçi. *Scalable Inference for Structured Gaussian Process Models.* PhD thesis, University of Cambridge, Cambridge, UK, 2012.

S. Särkkä and Á. F. García-Fernández. Temporal parallelization of Bayesian filters and smoothers. *arXiv preprint arXiv:1905.13002*, 5 2019.

S. Särkkä and A. Solin. *Applied Stochastic Differential Equations*. Institute of Mathematical Statistics Textbooks. Cambridge University Press, 2019.

S. Särkkä, A. Solin, and J. Hartikainen. Spatiotemporal learning via infinite-dimensional Bayesian filtering and smoothing. *IEEE Signal Processing Magazine*, 30(4):51–61, 2013.

A. Solin, J. Hensman, and R. E. Turner. Infinite-horizon Gaussian processes. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

K. E. Taylor, R. J. Stouffer, and G. A. Meehl. An overview of CMIP5 and the experiment design. *Bulletin of the American Meteorological Society*, 93(4): 485–498, 2012.

Y. W. Teh and M. Seeger. Semiparametric latent factor models. In *International Workshop on Artificial Intelligence and Statistics*, volume 10, 2005.

M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 61(3):611–622, 1999.

M. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pages 567–574. PMLR, 2009.

S. T. Tokdar and J. K. Ghosh. Posterior consistency of logistic Gaussian process priors in density estimation. *Journal of Statistical Planning and Inference*, 137(1): 34–42, 2007.

K. Ulrich, D. E. Carlson, K. Dzirasa, and L. Carin. GP kernels for cross-spectrum analysis, 2015.

H. Wackernagel. *Multivariate Geostatistics*. Springer-Verlag Berlin Heidelberg, 3 edition, 2003.

A. Wilson and H. Nickisch. Kernel interpolation for scalable structured Gaussian processes (KISS-GP). 37:1775–1784, 2015.

A. G. Wilson, D. A. Knowles, and Z. Ghahramani. Gaussian process regression networks. In *International Conference on Machine Learning*, volume 29. Omnipress, 2012.

T. Wu, L. Song, W. Li, Z. Wang, H. Zhang, X. Xin, Y. Zhang, L. Zhang, J. Li, F. Wu, et al. An overview of bcc climate system model development and application for climate change studies. *Journal of Meteorological Research*, 28(1):34–56, 2014.

B. M. Yu, J. P. Cunningham, G. Santhanam, S. I. Ryu, K. V. Shenoy, and M. Sahani. Gaussian-Process factor analysis for low-dimensional single-trial analysis of neural population activity. In *Advances in Neural Information Processing Systems*, volume 21, pages 1881–1888. Curran Associates, Inc., 2009.

Supplementary Material: Scalable Exact Inference in Multi-Output Gaussian Processes

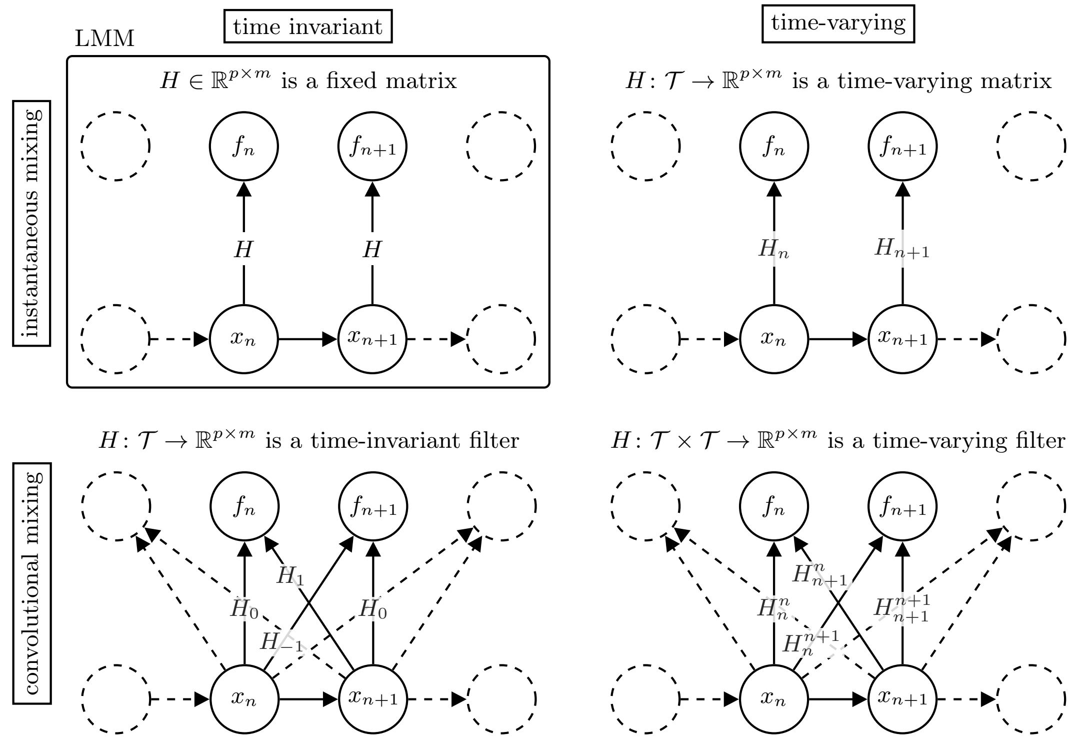

time invariant
time-varying
LMM
$H \in \mathbb{R}^{p\times m}$ is a fixed matrix
$H\colon \mathcal{T} \to \mathbb{R}^{p\times m}$ is a time-varying matrix
instantaneous mixing
f_n
f_{n+1}
H
H
H_n
H_{n+1}
x_n
x_{n+1}
$H\colon \mathcal{T} \to \mathbb{R}^{p\times m}$ is a time-invariant filter
$H\colon \mathcal{T} \times \mathcal{T} \to \mathbb{R}^{p\times m}$ is a time-varying filter
convolutional mixing
H_1
H_0
H_0
H_{-1}
H^n_{n+1}
H^n_n
H^{n+1}_{n+1}
H^{n+1}_n

Figure 6: Graphical models illustrating the difference between time-invariant/time-varying and instantaneous/convolutional multi-output GP models, for data sampled at real-valued times $t_1, t_2, \ldots$ (sampling period Δt). Abbreviations used: $x_n = x(t_n)$, $f_n = f(t_n)$, $H_n = H(n\Delta t)$, and $H^m_n = H(t_m, t_n)$. For simplicity, the dynamics of x are depicted as a Markov chain; since x is modelled with a GP, x_n actually depends on $x_{n'}$ for all $n' \le n$.

A Characterisation of MOGPs

Figure 6 illustrates the graphical models of time-invariant/time-varying and instantaneous/convolutional multi-output GP models.

B Maximum Likelihood Estimate

Throughout the appendix, we assume that the columns of H are linearly independent and $\Lambda > 0$. As a consequence, $H^\top \Lambda^{-1} H$ is invertible and $H^\top \Lambda^{-1} H > 0$.

Prop. 3. Denote $p(y \,|\, x) = \mathcal{N}(y; Hx, \Lambda)$, and let T be the $m \times p$ matrix $(H^\top \Lambda^{-1} H)^{-1} H^\top \Lambda^{-1}$. Then

$$Ty = \underset{x}{\arg\max}\; p(y \,|\, x)$$

and Ty is an unbiased estimate of x: $\mathbb{E}[Ty \,|\, x] = x$.

Proof. Note that

$$\log p(y \,|\, x) \simeq -\tfrac{1}{2}(y - Hx)^\top \Lambda^{-1}(y - Hx)$$

Using invertibility of $H^\top \Lambda^{-1} H$, an elementary calculation then shows that the maximum with respect to x is given by

$$x = (H^\top \Lambda^{-1} H)^{-1} H^\top \Lambda^{-1} y = Ty.$$

To show that Ty is an unbiased estimate of x, we use that $\mathbb{E}[y \,|\, x] = Hx$:

$$\mathbb{E}[Ty \,|\, x] = THx = (H^\top \Lambda^{-1} H)^{-1}(H^\top \Lambda^{-1} H)x = x.$$

□

C Proof of Prop. 1

To prove Prop. 1, we need the property of T that it "preserves the signal-to-noise ratio". This is characterised in the following lemma.

Lem. 1.

$$\frac{\mathcal{N}(y; Hx, \Lambda)}{\mathcal{N}(y; 0, \Lambda)} = \frac{\mathcal{N}(Ty; x, (H^\top \Lambda^{-1} H)^{-1})}{\mathcal{N}(Ty; 0, (H^\top \Lambda^{-1} H)^{-1})}.$$

Proof. It is simple to check the equality by direct verification. We show, however, how the equality may be derived. To begin with, we have

$$(y - Hx)^\top \Lambda^{-1}(y - Hx) = y^\top \Lambda^{-1} y - 2x^\top H^\top \Lambda^{-1} y + x^\top H^\top \Lambda^{-1} Hx.$$

Here

$$H^\top \Lambda^{-1} y = (H^\top \Lambda^{-1} H)(H^\top \Lambda^{-1} H)^{-1} H^\top \Lambda^{-1} y = (H^\top \Lambda^{-1} H)Ty,$$

so

$$(y - Hx)^\top \Lambda^{-1}(y - Hx) = y^\top \Lambda^{-1} y - 2x^\top (H^\top \Lambda^{-1} H)Ty + x^\top (H^\top \Lambda^{-1} H)x.$$

Adding and subtracting $yT^\top (H^\top \Lambda^{-1} H)Ty$, we find

$$(y - Hx)^\top \Lambda^{-1}(y - Hx) = y^\top \Lambda^{-1} y - yT^\top (H^\top \Lambda^{-1} H)Ty + (x - Ty)^\top (H^\top \Lambda^{-1} H)(x - Ty).$$

Hence, rearranging,

$$(y - Hx)^\top \Lambda^{-1}(y - Hx) - y^\top \Lambda^{-1} y = (x - Ty)^\top (H^\top \Lambda^{-1} H)(x - Ty) - yT^\top (H^\top \Lambda^{-1} H)Ty,$$

which yields the result. □

Proof (Prop. 1). By a general characterisation of minimal sufficient statistics (see, e.g., Th. 6.2.13 in Casella and Berger, 2001), Ty is a minimal sufficient statistic for x if and only if it is true that $p(y_1 \,|\, x)/p(y_2 \,|\, x)$ is constant as a function of x if and only if $Ty_1 = Ty_2$. Indeed, by Lem. 1,

$$\log \frac{p(y_1 \,|\, x)}{p(y_2 \,|\, x)} = (Ty_1 - Ty_2)^\top (H^\top \Lambda^{-1} H)^{-1} x + \text{const.}$$

which, by invertibility of $H^\top \Lambda^{-1} H$, does not depend on x if and only if $Ty_1 = Ty_2$. □

D Characterisation of Diagonal Projected Noise

Lem. 2. Let U be a truncated orthogonal matrix and $D > 0$ diagonal. Then every solution $M > 0$ of $U^\top M^{-1} U = D^{-1}$ is of the form $M = UDU^\top + VEV^\top$ with $E > 0$ diagonal and V such that $\begin{bmatrix} U & V \end{bmatrix}$ is orthogonal.

Proof. We equivalently show that M^{-1} for every solution $M > 0$ of $U^\mathsf{T} M U = D^{-1}$ is of the desired form. Denote

$$\mathcal{D} = \{M \geq 0 : U^\mathsf{T} M U = D^{-1}\}, \quad \mathcal{N} = \{M \geq 0 : U^\mathsf{T} M U = 0\}.$$

If $M_1, M_2 \in \mathcal{D}$, then $U^\mathsf{T}(M_1 - M_2)U = 0$ shows that $M_1 - M_2 \in \mathcal{N}$. Therefore, if $M \in \mathcal{D}$, then $\mathcal{D} = M + \mathcal{N}$. It is clear that $UD^{-1}U^\mathsf{T} \in \mathcal{D}$. Thus, $\mathcal{D} = UD^{-1}U^\mathsf{T} + \mathcal{N}$. Let $M \in \mathcal{N}$, and let $M = VEV^\mathsf{T}$ be the spectral decomposition of M. Assume that $E > 0$ by dropping the appropriate columns from V. Then $U^\mathsf{T} M U = 0$ says that U and V have pairwise orthogonal columns, so the columns of V are in $U^\perp$. We conclude that every $M \in \mathcal{D}$ is of the form $M = UD^{-1}U^\mathsf{T} + VEV^\mathsf{T}$ with $E > 0$ diagonal and V a truncated orthogonal matrix with columns in $U^\perp$. If $M > 0$, then in particular M has full rank, so it is clear that $\begin{bmatrix} U \; V \end{bmatrix}$ must form an orthogonal matrix. Therefore, $M^{-1} = UDU^\mathsf{T} + VE^{-1}V^\mathsf{T}$, which is of the desired form. □

Prop. 4. The projected noise is diagonal if and only if H is of the form $H = \Lambda^{\frac{1}{2}} U S^{\frac{1}{2}}$ with U a truncated orthogonal matrix and $S > 0$ diagonal. Suppose that this is the case, and fix such a U. Then H is of the form $H = UD^{\frac{1}{2}}$ with $D > 0$ diagonal if and only if every column of U is an eigenvector of Λ.

Proof. The projected noise is diagonal if and only if $H^\mathsf{T} \Lambda^{-1} H = S$ for some $S > 0$ diagonal. This condition is equivalent to

$$S^{-\frac{1}{2}} H^\mathsf{T} \Lambda^{-\frac{1}{2}} \Lambda^{-\frac{1}{2}} H S^{-\frac{1}{2}} = I,$$

which, in turn, holds if and only if $\Lambda^{-\frac{1}{2}} H S^{-\frac{1}{2}} = U$ is a truncated orthogonal matrix. Thus, the projected noise is diagonal if and only if H is of the form $H = \Lambda^{\frac{1}{2}} U S^{\frac{1}{2}}$ with U a truncated orthogonal matrix and $S > 0$ diagonal.

For the second statement, note that the condition that every column of U is an eigenvector of Λ is equivalent to the condition that Λ is of the form $\Lambda = UDU^\mathsf{T} + VEV^\mathsf{T}$ with $D, E > 0$ diagonal and V such that $\begin{bmatrix} U \; V \end{bmatrix}$ forms an orthogonal matrix. Suppose that Λ is of this form. Then

$$\Lambda^{\frac{1}{2}} = UD^{\frac{1}{2}}U^\mathsf{T} + VE^{\frac{1}{2}}V^\mathsf{T}$$

implies that

$$H = UD^{\frac{1}{2}}U^\mathsf{T}US^{\frac{1}{2}} + VE^{\frac{1}{2}}V^\mathsf{T}US^{\frac{1}{2}} = UD^{\frac{1}{2}}S^{\frac{1}{2}},$$

which is of the desired form. Conversely, suppose that $H = UD^{\frac{1}{2}}$ with $D > 0$ diagonal. Then

$$S^{-\frac{1}{2}} H^\mathsf{T} \Lambda^{-1} H S^{-\frac{1}{2}} = I = S^{-\frac{1}{2}} D^{\frac{1}{2}} U^\mathsf{T} \Lambda^{-1} U D^{\frac{1}{2}} S^{-\frac{1}{2}} \implies U^\mathsf{T} \Lambda^{-1} U = SD^{-1}.$$

Therefore, by Lem. 2,

$$\Lambda = UD^{-1}SU^\mathsf{T} + VEV^\mathsf{T}$$

with $E > 0$ diagonal and V such that $\begin{bmatrix} U \; V \end{bmatrix}$ is orthogonal, which again is of the desired form. □

E Form of Projection and Projected Noise

Prop. 5. Let H and Λ be of the form for the OLMM (Mod. 2). Then

$$T = S^{-\frac{1}{2}} U^\mathsf{T} \quad \text{and} \quad \Lambda_T = \sigma^2 S^{-1} + D.$$

Proof. To begin with, using the matrix inversion lemma once and once in reverse,

$$\begin{aligned} H^\mathsf{T} \Lambda^{-1} H &= S^{\frac{1}{2}} U^\mathsf{T} (\sigma^2 I + U S^{\frac{1}{2}} D S^{\frac{1}{2}} U^\mathsf{T})^{-1} U S^{\frac{1}{2}} \\ &= S^{\frac{1}{2}} U^\mathsf{T} (\sigma^{-2} I - \sigma^{-4} U S^{\frac{1}{2}} (D^{-1} + S^{\frac{1}{2}} \sigma^{-2} U^\mathsf{T} U S^{\frac{1}{2}})^{-1} S^{\frac{1}{2}} U^\mathsf{T}) U S^{\frac{1}{2}} \\ &= \sigma^{-2} S - \sigma^{-4} S (D^{-1} + \sigma^{-2} S)^{-1} S \\ &= (\sigma^2 S^{-1} + D)^{-1}, \end{aligned}$$

so

$$\Lambda_T = T\Lambda T^\top = (H^\top \Lambda^{-1} H)^{-1} = \sigma^2 S^{-1} + D.$$

Furthermore, by a similar calculation,

$$\begin{aligned}
T &= (H^\top \Lambda^{-1} H)^{-1} H^\top \Lambda^{-1} \\
&= (\sigma^2 S^{-1} + D) S^{\frac{1}{2}} U^\top (\sigma^{-2} I - \sigma^{-4} U S^{\frac{1}{2}} (D^{-1} + \sigma^{-2} S)^{-1} S^{\frac{1}{2}} U^\top) \\
&= (\sigma^2 S^{-1} + D)(\sigma^{-2} S - \sigma^{-4} S (D^{-1} + \sigma^{-2} S)^{-1} S) S^{-\frac{1}{2}} U^\top \\
&= (\sigma^2 S^{-1} + D)(\sigma^2 S^{-1} + D)^{-1} S^{-\frac{1}{2}} U^\top \\
&= S^{-\frac{1}{2}} U^\top.
\end{aligned}$$

□

F Decomposition of the Mean Squared Error

Prop. 6. Let $H = US^{\frac{1}{2}}$ with U a truncated orthogonal matrix and $S^{\frac{1}{2}} > 0$ diagonal. Then

$$\underbrace{\|y - Hx\|^2}_{\text{MSE}} = \underbrace{\|P_{U^\perp} y\|^2}_{\substack{\text{variance not}\\ \text{captured by basis}}} + \sum_{i=1}^{m} \overbrace{S_{ii}}^{\substack{\text{variance of}\\ i^{\text{th}}\text{ latent process}}} \underbrace{((Ty)_i - x_i)^2}_{\substack{\text{MSE of}\\ i^{\text{th}}\text{ latent process}}}$$

where $T = S^{-\frac{1}{2}} U^\top$ and $P_{U^\perp}$ is the orthogonal projection onto the orthogonal complement of U.

Proof. By expanding and using orthogonality of U,

$$\begin{aligned}
\|y - Hx\|^2 &= \|y\|^2 - 2\langle y, US^{\frac{1}{2}} x\rangle + \|US^{\frac{1}{2}} x\|^2 \\
&= \|y\|^2 - \|U^\top y\|^2 + \|U^\top y\|^2 - 2\langle U^\top y, S^{\frac{1}{2}} x\rangle + \|S^{\frac{1}{2}} x\|^2 \\
&= \langle y, (I - UU^\top) y\rangle + \sum_{i=1}^{m} (\langle u_i, y\rangle^2 - 2\langle u_i, y\rangle S_{ii}^{\frac{1}{2}} x_i + (S_{ii}^{\frac{1}{2}} x_i)^2) \\
&= \langle y, (I - UU^\top) y\rangle + \sum_{i=1}^{m} S_{ii} (S_{ii}^{-1} \langle u_i, y\rangle^2 - 2 S_{ii}^{-\frac{1}{2}} \langle u_i, y\rangle x_i + x_i^2) \\
&= \langle y, (I - UU^\top) y\rangle + \sum_{i=1}^{m} S_{ii} ((Ty)_i - x_i)^2,
\end{aligned}$$

where u_i is the i^{th} column of U. Note that $P_U = UU^\top$ is the orthogonal projection onto U, so $I - UU^\top = P_{U^\perp}$ is the orthogonal projection onto the orthogonal complement of U. Therefore,

$$\langle y, (I - UU^\top) y\rangle = \langle y, P_{U^\perp} y\rangle = \langle y, P_{U^\perp}^2 y\rangle = \langle P_{U^\perp}^\top y, P_{U^\perp} y\rangle = \langle P_{U^\perp} y, P_{U^\perp} y\rangle = \|P_{U^\perp} y\|^2.$$

□

G Kullback–Leibler Divergence Between an LMM and OLMM

Prop. 7. Consider two LMMs with equal $\Lambda = \sigma^2 I_p$, equal $K(t, t')$, but different bases H and $\hat{H}$. Let $t_1, \ldots, t_n \in \mathcal{T}$ and denote $x_i = x(t_i)$ and $y_i = y(t_i)$. It then holds that

$$\mathrm{D}_{\mathrm{KL}}(p(y_{1:n}, x_{1:n}) \,\|\, \hat{p}(y_{1:n}, x_{1:n})) = \mathrm{D}_{\mathrm{KL}}(\hat{p}(y_{1:n}, x_{1:n}) \,\|\, p(y_{1:n}, x_{1:n})) = n \frac{1}{2\sigma^2} \|H - \hat{H}\|_F^2$$

and

$$\inf_{\hat{H}:\text{OLMM}} \mathrm{D}_{\mathrm{KL}}(p(y_{1:n}, x_{1:n}) \,\|\, \hat{p}(y_{1:n}, x_{1:n})) \leq n\frac{\mathbb{E}[\|f(t)\|^2]}{\sigma^2} \max_i\,(1 - V_{ii}) \leq n\frac{\mathbb{E}[\|f(t)\|^2]}{2\sigma^2}\|I - V\|_F^2$$

where $\hat{H}$ ranges over matrices of the form $US^{\frac{1}{2}}$ with U a truncated orthogonal matrix and $S^{\frac{1}{2}} > 0$ diagonal, V is the orthogonal matrix collecting the right singular vectors of H, and $\mathbb{E}[\|f(t)\|^2]$ denotes the variance of the observations under the first LMM before adding noise.

Proof. Start out by expanding the Kullback–Leibler divergence and noting that $p(x_{1:n}) = \hat{p}(x_{1:n})$:

$$\begin{aligned}
\mathrm{D}_{\mathrm{KL}}(p(y_{1:n}, x_{1:n}) \,\|\, \hat{p}(y_{1:n}, x_{1:n})) &= -\mathbb{E}_{p(y_{1:n}, x_{1:n})} \log \frac{\hat{p}(y_{1:n} \,|\, x_{1:n})\cancel{\hat{p}(x_{1:n})}}{p(y_{1:n} \,|\, x_{1:n})\cancel{p(x_{1:n})}} \\
&= -\sum_{i=1}^{n} \mathbb{E}_{p(y_i, x_i)}[\log \hat{p}(y_i \,|\, x_i) - \log p(y_i \,|\, x_i)] \\
&= -\sum_{i=1}^{n} \mathbb{E}_{p(y_i, x_i)}[\log \mathcal{N}(y_i; \hat{H}x_i, \sigma^2 I_p) - \log \mathcal{N}(y_i; Hx_i, \sigma^2 I_p)].
\end{aligned}$$

Here

$$\mathbb{E}_{p(y_i, x_i)}[\log \mathcal{N}(y_i; \hat{H}x_i, \sigma^2 I_p)] = -\frac{p}{2}\log 2\pi\sigma^2 - \frac{1}{2\sigma^2}\mathbb{E}_{p(y_i, x_i)}[\|y_i - \hat{H}x_i\|^2]$$

where

$$\begin{aligned}
\mathbb{E}_{p(y_i, x_i)}[\|y_i - \hat{H}x_i\|^2] &= \mathbb{E}_{p(y_i, x_i)} \operatorname{tr}[y_i y_i^\top - 2y_i x_i^\top \hat{H}^\top + x_i x_i^\top \hat{H}\hat{H}^\top] \\
&= \operatorname{tr}[HH^\top + \sigma^2 I - 2H\hat{H}^\top + \hat{H}\hat{H}^\top] \\
&= p\sigma^2 + \operatorname{tr}[(H - \hat{H})(H - \hat{H})^\top] \\
&= p\sigma^2 + \|H - \hat{H}\|_F^2.
\end{aligned}$$

Therefore,

$$\mathrm{D}_{\mathrm{KL}}(p(y_{1:n}, x_{1:n}) \,\|\, \hat{p}(y_{1:n}, x_{1:n})) = n\frac{1}{2\sigma^2}\|H - \hat{H}\|_F^2.$$

Let $H = USV^\top$ be the SVD of H where U is a truncated orthogonal matrix with the same shape as H, $S > 0$ is a square diagonal matrix, and V is a square orthogonal matrix. Note that $U^\top U = I$, but $UU^\top \neq I$. Then, choosing $\hat{H} = US$,

$$\inf_{\hat{H}:\text{OLMM}} \mathrm{D}_{\mathrm{KL}}(p(y_{1:n}, x_{1:n}) \,\|\, \hat{p}(y_{1:n}, x_{1:n})) \leq n\frac{1}{2\sigma^2}\|U(SV^\top - S)\|_F^2 = n\frac{1}{2\sigma^2}\|SV^\top - S\|_F^2$$

since $\|UA\|_F^2 = \operatorname{tr}[A^\top U^\top UA] = \operatorname{tr}[A^\top A] = \|A\|_F^2$. We now further simplify:

$$\|SV^\top - S\|_F^2 = \operatorname{tr}[(SV^\top - S)(SV^\top - S)^\top] = \operatorname{tr}[SV^\top VS - SV^\top S - SVS + SS] = 2\operatorname{tr}[SS - SVS].$$

Hence, by definition of the trace and the fact that S is diagonal,

$$\|SV^\top - S\|_F^2 = 2\sum_{i=1}^{m} S_{ii}^2(1 - V_{ii}) \leq 2\left(\sum_{i=1}^{m} S_{ii}^2\right) \max_i\,(1 - V_{ii}) = 2\mathbb{E}[\|f\|^2] \max_i\,(1 - V_{ii}),$$

since

$$\mathbb{E}[\|f(t)\|^2] = \mathbb{E}\operatorname{tr}[f(t)f^\top(t)] = \operatorname{tr}[HH^\top] = \operatorname{tr}[S^2].$$

Therefore,

$$\|SV^\top - S\|_F^2 \leq 2\mathbb{E}[\|f\|^2] \max_i\,(1 - V_{ii}) \leq 2\mathbb{E}[\|f\|^2] \sum_{i=1}^{m}(1 - V_{ii}) = \mathbb{E}[\|f\|^2]\|I - V\|_F^2,$$

where the equality follows from a similar calculation:

$$\|I - V\|_F^2 = \operatorname{tr}[I - V^\top - V + V^\top V] = 2\operatorname{tr}[I - V].$$

□

H Missing Data

Missing data is troublesome for the OLMM, because it is not possible to take away a subset of the rows of H and retain orthogonality of the columns. Suppose that data are missing for a fixed subset of outputs. Collect that subset of outputs into y_1, and let y_2 be the remainder of the outputs. The issue can then be resolved by considering a *multi-headed* version of the OLMM:

$$y_1 \,|\, H_1, x \sim \mathcal{GP}(H_1 x(t), \delta[t-t']\Lambda_1),$$
$$y_2 \,|\, H_2, x \sim \mathcal{GP}(H_2 x(t), \delta[t-t']\Lambda_2).$$

For k heads, if $y_I = (y_i)_{i \in I}$ for a subset of heads $I \subseteq \{1, \dots, k\}$ are observed, then

$$T_I y_I = \Lambda_{T_I} \textstyle\sum_{i \in I} \Lambda_{T_i}^{-1} T_i y_i, \quad \Lambda_{T_I}^{-1} = \textstyle\sum_{i \in I} \Lambda_{T_i}^{-1}.$$

While this new model resolves the missing data problem, the assumption that each collection of outputs (each head) is a linear transformation of the same collection of latent processes will be overly restrictive for many contexts: it is not typically the case that our prior beliefs about the relationships between outputs have anything to do with the availability of data pertaining to them.

In the general case, where data can be missing arbitrarily, let Y_{o} be the observed data. Complement Y_{o} with missing data Y_{m} such that $Y = Y_{\mathrm{o}} \cup Y_{\mathrm{m}}$ is complete. Then a simple solution is to use variational inference. Assume a Gaussian approximate posterior distribution $q(Y_{\mathrm{m}})$ over Y_{m}, and maximise the evidence lower bound (ELBO) $\mathcal{L}$ using gradient-based optimisation:

$$\log p(Y_{\mathrm{o}}) \geq \mathbb{E}_{q(Y_{\mathrm{m}})}[\log p(Y)] + H[q(Y_{\mathrm{m}})] = \mathcal{L}[q(Y_{\mathrm{m}})],$$

where the expectation can be approximated using the reparametrisation trick (Kingma and Welling, 2013), $\log p(Y)$ can be computed efficiently because Y is complete, and $H[q(Y_{\mathrm{m}})]$ denotes the entropy of $q(Y_{\mathrm{m}})$. This approach provides a tractable solution when the number of missing data are not too numerous.

I Heterogeneous Observation Noise

Although the specification of the observation noise $\Lambda = \sigma^2 I_p + HDH^\top$ in the OLMM does not allow for heterogeneous observation noise, it is possible to set $\Lambda = \operatorname{diag}(\sigma_1^2, \dots, \sigma_p^2)$ and use Prop. 4 to include Λ in the parametrisation of H: $H = \Lambda^{\frac{1}{2}} U S^{\frac{1}{2}}$. This parametrisation can be interpreted in two ways: *(i)* The model has a whitening transform built in. In the projection T, the (noise in the) data will first by whitened by $\Lambda^{-\frac{1}{2}}$. Hence, this parametrisation can be used as a more principled substitute for the usual data normalisation where the outputs are divided by their empirical standard deviation prior to feeding them to the model. *(ii)* The basis is orthogonal with respect to a weighted Euclidean inner product: $\langle h_i, h_j \rangle_{\Lambda^{-1}} = \sum_{k=1}^p h_{ik} h_{jk} / \sigma_k^2 = 0$ for $i \neq j$. Intuitively, this means that the basis is orthogonal in the usual sense after stretching the i^{th} dimension by σ_i^{-1}. Although this construction provides additional flexiblity, it does require that $D = 0$ to avoid a circular dependency between Λ and H.

J Point Process Experimental Details and Additional Results

We use a total of $np = 200 \times 100$ bins in the model, where each bin is $5\,\mathrm{m} \times 5\,\mathrm{m}$. The likelihood for this experiment is given by

$$p(Y \mid f) \approx \prod_{i=1}^{n} \prod_{j=1}^{p} \operatorname{Poisson}(Y_{ij} \mid \exp(f(r_{ij})))$$

where r_{ij} is the coordinate of the ij^{th} bin, Y_{ij} is the number of data points in the ij^{th} bin, and Y is the $n \times p$ matrix of counts. We perform 10^5 iterations of Gibbs sampling, alternating between Elliptical

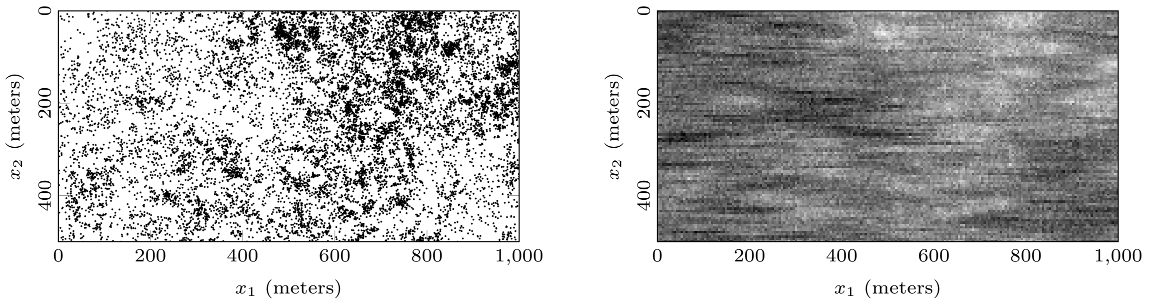


Figure 7: Observations of the rainforest tree locations (left), and posterior mean log-intensity for the log-Gaussian Cox process model (right) with a grid of $np = 20000$ observation bins.

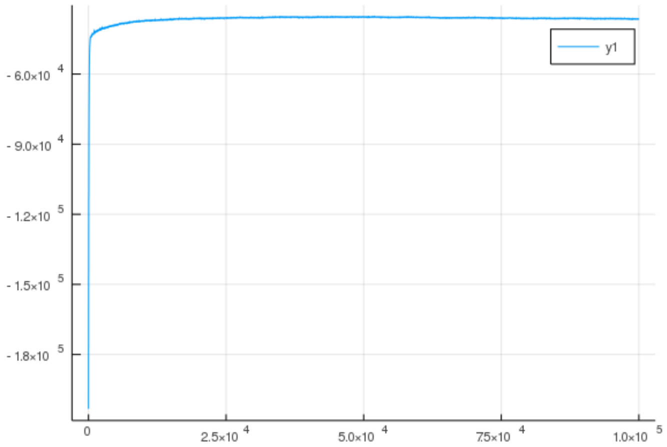


Figure 8: Log–joint probability per iteration

Slice Sampling for the Gaussian process given its hyperparameters, and Metropolis Hastings with proposal distribution $\mathcal{N}(\theta, 0.05^2)$ for the log of the hyperparameters given the latent GP-distributed function. The kernel is a product of two Matérn-5/2 kernels, one over the vertical dimension and the other over the horizontal one, each with their own length scale. A single process variance is utilised, and a nugget term is added. The log of each of the four hyperparameters was given a $\mathcal{N}(0, 1)$ prior. Figure 8 shows the log joint of the entire state after each iteration, while Fig. 9 shows the progress of each hyperparameter per iteration.

The times in Fig. 4 were obtained via `BenchmarkTools.jl` (Chen and Revels, 2016). The implementation of the standard Kronecker product decomposition trick makes use of `Kronecker.jl`, and Julia's (Bezanson et al., 2017) standard linear algebra libraries, which make use of OpenBLAS and LAPACK to efficiently perform matrix-matrix products and compute eigendecompositions. The implementation of the state-space GP additionally makes use of `StaticArrays.jl` for efficient stack-allocated matrices, and `Stheno.jl` for GP-related functionality. Timing experiments were conducted on a single CPU core to make the implementations comparable.

The state-space implementation of the GP makes use of the trick introduced by Solin et al. (2018) when computing the log marginal likelihood. However, the infinite-horizon trick is exploited only once the filtering covariance has converged, which is determined by the point at which the Frobenius norm of the difference between the filtering covariance at the t^{th} and $(t-1)^{\text{th}}$ iterations drops below 10^{-12}. In practice, we find that this produces log-marginal likelihoods that are essentially exact, as expected.

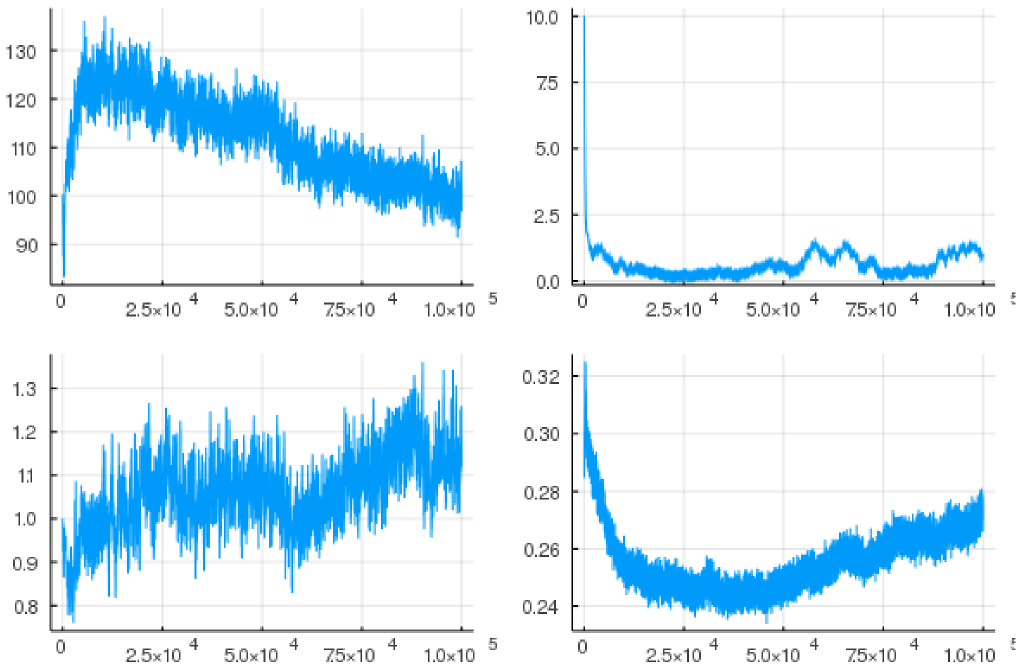


Figure 9: Hyperparameters per iteration. Shows the horizontal length scale (top-left), vertical length scale (top-right), nugget variance (bottom-right), and process variance (bottom-left).