Scalable Exact Inference in Multi-Output Gaussian Processes

Wessel P. Bruinsma [1 2] **Eric Perim** [2] **Will Tebbutt** [1] **J. Scott Hosking** [3 4] **Arno Solin** [5] **Richard E. Turner** [1 6]

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 their computational scaling $O(n^3p^3)$, which is cubic in the number of both inputs n (*e.g.*, time points or locations) and outputs p. For this reason, a popular class of MOGPs assumes that the data live around a low-dimensional linear subspace, reducing the complexity to $O(n^3m^3)$. However, this cost is still cubic in the dimensionality of the subspace m, which is still prohibitively expensive for many applications. We propose the use of a sufficient statistic of the data to accelerate inference and learning in MOGPs with orthogonal bases. The method achieves *linear* scaling in m in practice, allowing these models to scale to large m without sacrificing significant expressivity or requiring approximation. 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. We demonstrate the efficacy of the method on various synthetic and real-world data sets.

1. Introduction

Gaussian processes (GPs, Rasmussen & Williams, 2006) form an interpretable, modular, and tractable probabilistic framework for modelling nonlinear functions. They are successfully applied in a wide variety of single-output problems: 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 & 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).

Multi-output Gaussian processes (MOGPs) leverage the flexibility and interpretability of GPs while capturing structure across outputs. One of the first applications of GPs with multiple outputs was in geostatistics (Matheron, 1969). Today, MOGPs models can be found in various areas, including geostatistics (Wackernagel, 2003), factor analysis (Teh & Seeger, 2005; Yu et al., 2009), dependent or multi-task learning (Boyle & Frean, 2005; Bonilla et al., 2007; 2008; Osborne et al., 2008), latent force models (Álvarez et al., 2009; Álvarez & Lawrence, 2009; Álvarez et al., 2010; Álvarez & Lawrence, 2011), state space modelling (Särkkä et al., 2013), regression networks (Wilson et al., 2012; Nguyen & Bonilla, 2014; Dezfouli et al., 2017), and mixture models (Ulrich et al., 2015; Bruinsma, 2016; Parra & Tobar, 2017; Requeima et al., 2019).

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 & Rasmussen, 2005; Titsias, 2009; Lázaro-Gredilla et al., 2010; Hensman et al., 2013; Wilson & Nickisch, 2015; Bui et al., 2017; Cheng & Boots, 2017; Hensman et al., 2018). To mitigate these unfavourable scalings, a particular class of MOGPs, which we call the Instantaneous Linear Mixing Model (ILMM, Sec. 2.1), assumes that the data live around an m-dimensional linear subspace, where $m < p$. This class exploits the low-rank structure of its covariance to reduce the complexity of inference and learning to roughly $O(n^3m^3)$ time and $O(n^2m^2)$ memory. Although m is typically much smaller than p, the runtime complexity is again cubic in m. Consequently, the ILMM is prohibitively expensive in applications where moderate m is required. Consider, for example, hourly real-time electricity prices at 2313 different locations across 15 U.S. states and the Canadian province of Manitoba during the year 2019 (MISO, 2019). Forecasting electricity prices is crucial in the planning of energy transmission, which happens 24 hours in advance. The ILMM is particularly well suited to this task: the prices derive from optimal power flow, which tends to exhibit low-rank structure. However, it still takes roughly $m = 40$ to explain 95% of the variance of this data. For $n = 1\,\mathrm{k}$ time points, this

[1]University of Cambridge [2]Invenia Labs [3]British Antarctic Survey [4]Alan Turing Institute [5]Aalto University [6]Microsoft Research. Correspondence to: Wessel P. Bruinsma <wpb23@cam.ac.uk>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020.

requires the inversion of a $40\,\mathrm{k} \times 40\,\mathrm{k}$ matrix. Even worse, to explain 99% of the variance, it requires the inversion of a $120\,\mathrm{k} \times 120\,\mathrm{k}$ matrix, clearly far beyond what is feasible.

In this paper, we develop a new perspective on MOGPs in the Instantaneous Linear Mixing Model class through the use of a sufficient statistic of the data. We use this sufficient statistic to identify a class of MOGPs, which we call the Orthogonal Instantaneous Linear Mixing Model (OILMM, Sec. 3), in which inference and learning take $O(n^3m + nmp + m^2p)$ time and $O(n^2m + np + mp)$ memory, without sacrificing significant expressivity nor requiring any approximations. The dominant (first) terms in these expressions are *linear* in m, rather than cubic. It is this feature that allows the OILMM to scale to large m. The linear scaling is achieved by breaking down the high-dimensional multi-output problem into independent single-output problems, whilst retaining exact inference. Consequently, the proposed methodology is interpretable—*e.g.*, it can be seen as a natural generalisation of probabilistic principal component analysis (PPCA, Tipping & Bishop, 1999)—simple to implement, and trivially compatible with single-output scaling techniques in a plug-and-play way. For example, it can be combined with the variational inducing point approximation by (Titsias, 2009) or with state-space approximation methods (Sec. 3.9); these approximations reduce the time complexity of the dominant term to linear in *both* the number of data points n and m. We demonstrate the efficacy of the OILMM in experiments on various synthetic and real-world data sets. Simple algorithms to perform inference and learning in the OILMM are presented in App. A.

2. Multi-Output Gaussian Process 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 collection 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. We consider the input space time, where $\mathcal{T} = \mathbb{R}$, but the analysis trivially applies to more general feature spaces, *e.g.* $\mathcal{T} = \mathbb{R}^d$. A MOGP $f \sim \mathcal{GP}(m, K)$ is described by a *vector-valued* mean function $m(t) = \mathbb{E}[f(t)]$ and a *matrix-valued* covariance function $K(t, t') = \mathbb{E}[f(t)f^\mathsf{T}(t')] - \mathbb{E}[f(t)]\mathbb{E}[f^\mathsf{T}(t')]$. 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. The Instantaneous Linear Mixing Model

A simple, but general class of MOGPs decomposes a signal $f(t)$ comprising p outputs into a fixed basis $h_1, \ldots, h_m \in \mathbb{R}^p$ with coefficients $x_1(t), \ldots, x_m(t) \in \mathbb{R}$:

$$f(t) = h_1 x_1(t) + \ldots + h_m x_m(t) = Hx(t)$$

where h_i is the i^{th} column of H. The coefficients $x_1(t), \ldots, x_m(t)$ are time varying and modelled independently with unit-variance Gaussian processes. The noisy signal $y(t)$ is then generated by adding $\mathcal{N}(0, \Sigma)$-distributed noise to $f(t)$. Intuitively, this means that the p-dimensional data live in a "pancake" (Roweis & Ghahramani, 1999; MacKay, 2002) around the m-dimensional column space of H, where typically $m \ll p$.

Mod. 1 (Instantaneous 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 Σ a $p \times p$ observation noise covariance. Then the ILMM is given by the following generative model:

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

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

The ILMM is definitely not novel; the specific formulation in Mod. 1 is for convenience of the exposition in this paper. In particular, the ILMM is very similar to the Linear Model of Coregionalisation (LMC) (Goovaerts, 1997). In the LMC, every latent process has multiple independent copies and the observation noise Σ is typically diagonal. More generally, the ILMM is a special case of the more general formulation with mixing mechanism $f(t) = \int \tilde{H}(t, \tau)x(\tau)\,\mathrm{d}\tau$ where $\tilde{H} \colon \mathcal{T} \times \mathcal{T} \to \mathbb{R}^{p \times m}$ is a matrix-valued time-varying filter. In particular, it is the case $\tilde{H}(t, \tau) = \delta(t - \tau)H$; here the mixing is *instantaneous* and *time-invariant*. Many other MOGPs in the machine learning and geostatistics literature can be seen as specialisations of this more general formulation by imposing structure on $\tilde{H}$ and K. An organisation of the literature from this point of view, which we call the Mixing Model Hierarchy (MMH), is presented in App. B.

2.2. Inference and Learning in the ILMM

The complexities of inference and learning in MOGPs can often be alleviated by exploiting structure in the kernel. This is the case for the ILMM, which we have seen exhibits low-rank covariance structure. In this section, we develop a new perspective on the ILMM by showing that the covariance structure can be exploited by devising a low-dimensional "summary" or "projection" of the p-dimensional observations. This reduces the complexities from $O(n^3p^3)$ time and $O(n^2p^2)$ memory to $O(n^3m^3 + nmp + m^2p)$ time and $O(n^2m^2 + np + mp)$ memory, where $O(nmp + m^2p)$ is

the cost of projecting the data and computing the projection, and $O(np + mp)$ is the cost of storing the data and projection.

The low-dimensional projection of the observations y will be given by a sufficient statistic for the model, which is therefore "without loss of information" and can be used to accelerate inference. Concretely, the projection of y is given by the maximum likelihood estimate (MLE) of x under the likelihood $p(y \mid x)$ of the ILMM. As Prop. 2 in App. D shows, this MLE is given by Ty where T is the $m \times p$ matrix $(H^\mathsf{T}\Sigma^{-1}H)^{-1}H^\mathsf{T}\Sigma^{-1}$; Ty is an unbiased estimator of x. Because Ty is an MLE for x, it is a function of a sufficient statistic for x, if one exists. Prop. 3 in App. E shows that Ty is actually minimally sufficient itself. For any prior $p(x)$ over x, sufficiency of Ty gives that $p(x \mid y) = p(x \mid Ty)$; that is, 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 proven in App. F:

Prop. 1. 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 Σ a $p \times p$ observation noise covariance. Then consider the following generative model:

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

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

$$Ty \mid x \sim \mathcal{GP}(x(t), \delta[t-t']\Sigma_T) \text{ with } \Sigma_T = (H^\mathsf{T}\Sigma^{-1}H)^{-1}.$$

Moreover, the probability of the data Y is given by

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

where the i^{th} observation y_i is the i^{th} column of Y.

Crucially, Y are p-dimensional observations, TY are m-dimensional summaries, and typically $m \ll p$, so conditioning on TY is often much cheaper; note that computing TY takes $O(nmp)$ time and $O(mp)$ memory. In particular, if we apply Prop. 1 to the ILMM 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. 1a.

The case of Prop. 1 where x is Gaussian can be found as Results 1 and 2 by Higdon et al. (2008), and was also used by the authors to accelerate inference. Although the reduction in computational complexities allows Higdon et al. to scale to significantly larger data, they are still limited by the cubic dependency on m.

If the observations can be naturally represented as multi-index arrays in $\mathbb{R}^{p_1 \times \cdots \times p_q}$, a natural choice is to correspondingly decompose $H = H_1 \otimes \cdots \otimes H_q$ where $\otimes$ is the Kronecker product. In this parametrisation, the projection and projected noise also become the Kronecker products: $T = T_1 \otimes \cdots \otimes T_q$ and $\Sigma_T = \Sigma_{T_1} \otimes \cdots \otimes \Sigma_{T_q}$. See App. H. The model by Zhe et al. (2019) can be seen as an ILMM of this form with $K(t, t') = k(t, t')I_m$ where k is a scalar-valued kernel and $\Sigma = \sigma^2 I_p$.

In Prop. 1, we call $\Sigma_T = T\Sigma T^\mathsf{T} = (H^\mathsf{T}\Sigma^{-1}H)^{-1}$ the *projected observation noise*. The projected noise Σ_T is important, because it couples the latent processes upon observing data. In particular, if the latent processes are independent under the prior and Σ_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 Instantaneous Linear Mixing Model.

2.3. Interpretation of the Likelihood

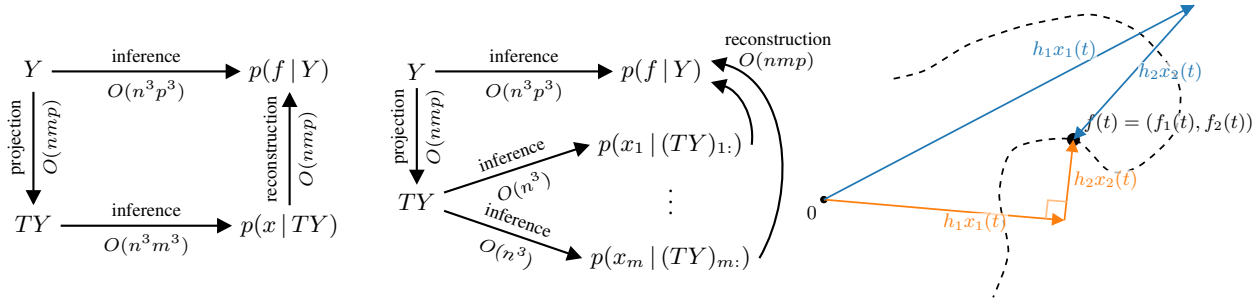
Prop. 1 shows that the log-probability of the data Y is equal to the log-probability of the projected data TY plus, for every observation y_i, a correction term of the form $\log \mathcal{N}(y_i \mid 0, \Sigma)/\mathcal{N}(Ty_i \mid 0, \Sigma_T)$. Prop. 4 in App. G shows that this correction term can be written as

$$-\tfrac{1}{2}(p-m)\log 2\pi - \underbrace{\tfrac{1}{2}\log|\Sigma|/|\Sigma_T|}_{\text{noise "lost by projection"}} - \underbrace{\tfrac{1}{2}\|y_i - HTy_i\|^2_\Sigma}_{\text{data "lost by projection"}},$$

where $\|\cdot\|_\Sigma = \|\Sigma^{-\frac{1}{2}}\cdot\|$. When the likelihood is optimised with respect to H, the correction terms will prevent the projection T from discarding a component of the data Y and the noise Σ that is "too large". For example, for the ILMM, if these correction terms were ignored, then after optimising we would find that $TY = 0$ and $\Sigma_T = 0$, because the density of a zero-mean Gaussian is highest at the origin, and becomes higher as the variance becomes smaller; it is exactly $TY = 0$ and $\Sigma_T = 0$ that the two penalties prevent from happening.

3. The Orthogonal Instantaneous Linear Mixing Model

Inspired by Prop. 1, we will now identify a subclass of the ILMM for which, in practice, inference and learning scale *linearly* in the number of latent processes m rather than cubically. As we will see, this happens when the projected observation noise is diagonal, which is the case for the Orthogonal Instantaneous Linear Mixing Model (OILMM): the subclass of ILMMs where the basis H is *orthogonal*. In particular, $H = US^{\frac{1}{2}}$ where U is a matrix with orthonormal



(a) Conditioning on Y in the ILMM (b) Conditioning on Y in the OILMM (c) Difference between the ILMM and OILMM

Figure 1. (a–b) Commutative diagrams depicting that conditioning on Y in the ILMM and OILMM is equivalent to conditioning respectively on TY and independently every x_i on $(TY)_{i:}$, but yield different computational complexities. The reconstruction costs assume computation of the marginals. (c) Illustration of the difference between the ILMM and OILMM. The trajectory of a particle (dashed line) in two dimensions is modelled by the ILMM (blue) and OILMM (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 OILMM, the basis vectors h_1 and h_2 are constrained to be orthogonal; in the ILMM, h_1 and h_2 are unconstrained.

columns and $S > 0$ a diagonal. We define this model as follows:

Mod. 2 (Orthogonal Instantaneous Linear Mixing Model)**.** The OILMM is an ILMM (Mod. 1) where the basis H is a $p \times m$ matrix of the form $H = US^{\frac{1}{2}}$ with U a matrix with orthonormal columns and $S > 0$ diagonal, and $\Sigma = \sigma^2 I_p + HDH^\mathsf{T}$ a $p \times p$ matrix with $D \geq 0$ diagonal.

The difference between the ILMM and the OILMM is illustrated in Fig. 1c. In the OILMM, we require that $m \leq p$, since the number of p-dimensional vectors that can be mutually orthogonal is at most p. Also, D in Σ can be interpreted as heterogeneous noise deriving from the latent processes. Moreover, although H and Σ do not depend on time, our analysis and results trivially carry over to the case where H_t and Σ_t do vary with time. Finally, for the OILMM, Prop. 8 in App. L shows that $T = S^{-\frac{1}{2}}U^\mathsf{T}$ and $\Sigma_T = \sigma^2 S^{-1} + D$.

Whereas the ILMM is a time-varying generalisation of FA, the OILMM can be seen as a time-varying generalisation of probabilistic principal component analysis (PPCA, Tipping & Bishop, 1999): $D = 0$ and $K(t, t') = \delta[t - t']I_m$ recovers the orthogonal solution of PPCA exactly; recall that PPCA admits infinitely many solutions, with only one corresponding to orthogonal axes, whereas the modelling assumptions of the OILMM recover this solution automatically. See Fig. 2 for a visualisation of the relationship between FA, PPCA, the ILMM, and the OILMM. The OILMM is also related 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 OILMM orthogonality of the columns of H is built into the model. In this respect, the OILMM is more similar to the model by Higdon et al. (2008), who also consider a MOGP with an orthogonal basis built in.

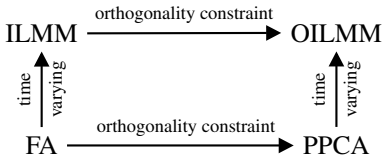


Figure 2. Relationship between factor analysis (FA), probabilistic principal component analysis (PPCA, Tipping & Bishop, 1999), the ILMM (Mod. 1), and the OILMM (Mod. 2)

3.1. Generality of the OILMM

A central theme of the experiments will be to assess how restrictive the orthogonality assumption is for the OILMM. In this section, we theoretically investigate this question from various perspectives. In the separable case, where $K(t, t') = k(t, t')I_m$ for a scalar-valued kernel k, for every ILMM with homogeneous observation noise ($\Sigma = \sigma^2 I_p$), there exists an OILMM with $D = 0$ that is equal in distribution to y. To see this, note

$$y^{(\text{ILMM})} \sim \mathcal{GP}(0, k(t, t')HH^\mathsf{T} + \sigma^2\delta[t - t']I_p),$$
$$y^{(\text{OILMM})} \sim \mathcal{GP}(0, k(t, t')USU^\mathsf{T} + \sigma^2\delta[t - t']I_p).$$

Hence, letting USU^T be the eigendecomposition of HH^T gives an OILMM equal in distribution to y. In the non-separable 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 OILMM, 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, Prop. 7 in App. K shows that the Kullback–Leibler (KL) divergence between two ILMMs with bases H and $\hat{H}$ is proportional to $\|H - \hat{H}\|_F^2$, hence symmetric, where $\|\cdot\|_F$ denotes the Frobenius norm. As a consequence (Prop. 7), the KL between an ILMM with basis H and the OILMM closest in

KL is upper bounded by $\|I_m - V\|_F^2$ where V are the right singular vectors of H. This makes sense: $V = I_m$ implies that H is of the form $US^{\frac{1}{2}}$ with U a matrix with orthonormal columns and $S > 0$ diagonal. It also shows that an ILMM is close to an OILMM if V is close to I_m in the sense of the Frobenius norm.

3.2. Choice of Basis

The basis H is a parameter of the model that can be learned through gradient-based optimisation of the likelihood. Parametrising the orthogonal part U of the basis H takes $O(m^2 p)$ time and $O(mp)$ memory (see App. C). This complexity is *quadratic* in m, rather than linear. However, the cost of parametrising U is typically far from dominant, which means that this cost is typically negligible. See App. I for a more detailed discussion.

Observing that $\mathbb{E}[f(t)f^\mathsf{T}(t)] = HH^\mathsf{T}$, a sensible initialisation of the basis H is (a truncation of) $\hat{U}\hat{S}^{\frac{1}{2}}$ where $\hat{\Sigma} = \hat{U}\hat{S}\hat{U}^\mathsf{T}$ is the eigendecomposition of an estimate $\hat{\Sigma}$ of the spatial covariance. In the case that there is a kernel over the outputs, *e.g.* in separable spatio-temporal GP models, H can be set to (a truncation of) $US^{\frac{1}{2}}$ where USU^T 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. See Sec. 3.9.

3.3. Diagonal Projected Noise

As alluded to in Sec. 2.1, under the OILMM, the projected noise Σ_T from Prop. 1 is diagonal: $\Sigma_T = \sigma^2 S^{-1} + D$; Prop. 6 in the App. J characterises exactly when this is the case. This property is crucial, because, as we explain in the next paragraph, it allows the model to break down the high-dimensional multi-output problem into independent single-output problems, which brings significant computational advantages.

3.4. 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 $(TY)_{i:} = Y^\mathsf{T} U_{i:}/\sqrt{S_{ii}}$ under noise $(\Sigma_T)_{ii} = \sigma^2/S_{ii} + D_{ii}$, which means that the high-dimensional prediction problem breaks down into independent single-output problems. Therefore, inference takes $O(n^3 m + nmp)$ time and $O(n^2 m + np)$ memory (see App. C), which are *linear* in m. This decoupled inference procedure is depicted in Fig. 1b and outlined in more detail in Apps. A.2 and A.3. Note that the decoupled problems can be treated in parallel to achieve sublinear wall time, and that in the separable case further speedups are possible.

3.5. Learning

For computing the marginal likelihood, the OILMM also offers computational benefits. Prop. 9 in App. M shows that $\log p(Y)$ from Prop. 3 simplifies to:

$$\begin{aligned}&\log p(Y)\\ &= -\frac{n}{2}\log|S| - \frac{n(p-m)}{2}\log 2\pi\sigma^2 - \frac{1}{2\sigma^2}\|(I_p - UU^\mathsf{T})Y\|_F\\ &\quad + \textstyle\sum_{i=1}^m \log \mathcal{N}((TY)_{i:} \,|\, 0, K_i + (\sigma^2/S_{ii} + D_{ii})I_n)\end{aligned}$$

where $\|\bullet\|_F$ denotes the Frobenius norm and K_i is the $n \times n$ kernel matrix for the i^{th} latent process x_i. We conclude that learning also takes $O(n^3 m + nmp)$ time and $O(n^2 m + np)$ memory (see App. C), again *linear* in the number of latent processes. Computation of the marginal likelihood is outlined in more detail in App. A.4.

3.6. Interpretability

Besides computational benefits, the fact that the OILMM breaks down into independent problems for the latent processes also promotes interpretability.[1] 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. For example, the OILMM allows a simple and interpretable decomposition of the mean squared error:

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

where $P_{H^\perp}$ is the orthogonal projection onto the orthogonal complement of $\mathrm{col}(H)$. See Prop. 10 in App. N for a proof.

3.7. 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 $O(nmr^2)$ time and $O(nmr)$ memory, ignoring the cost of the projection (see App. C). Most importantly, if $k(t, t')$ is Markovian (*e.g.* of the Matérn class), then one can leverage state-space methods to efficiently solve the m independent problems exactly (Hartikainen & Särkkä, 2010; Särkkä & Solin, 2019). This brings down the scaling to $O(nmd^3)$ time and $O(nmd^2)$ memory, where d is the state

[1] In the OILMM, the latent processes retain independence in the posterior distribution, which is not generally true for the ILMM.

dimension, typically $d \ll m, n$ (see App. C). We further discuss this approach in Sec. 3.9.

3.8. Missing Data

Missing data is troublesome for the OILMM, because it is not possible to take away a subset of the rows of H and retain orthogonality of the columns. In this section, we develop an approximation for the OILMM to deal with missing data in a simple and effective way. For a matrix or vector A, let A_o and A_m denote the rows of A corresponding to respectively observed and missing values. Also, for a matrix A, let $d[A]$ denote the diagonal matrix resulting from setting the off-diagonal entries of A to zero. In the case of missing data, Prop. 11 in App. O.1 shows that the projection and projected noise are given by $T_\mathrm{o} = S^{-\frac{1}{2}} {U_\mathrm{o}}^\dagger$ and $\Sigma_{T_\mathrm{o}} = \sigma^2 S^{-\frac{1}{2}} ({U_\mathrm{o}}^\mathsf{T} U_\mathrm{o})^{-1} S^{-\frac{1}{2}} + D$. Observe that Σ_{T_o} is dense, because, unlike U, the columns of U_o are not orthogonal. However, they may be approximately orthogonal, which motivates the approximation $\Sigma_{T_\mathrm{o}} \approx d[\Sigma_{T_\mathrm{o}}]$. Prop. 12 in App. O.1 shows that this approximation will be accurate if missing observations cannot decrease the norm of a vector in $\mathrm{col}(H)$ too much:

$$\varepsilon_\mathrm{rel} = \frac{\|\Sigma_{T_\mathrm{o}} - d[\Sigma_{T_\mathrm{o}}]\|_\mathrm{op}}{\|d[\Sigma_{T_\mathrm{o}}]\|_\mathrm{op}} \lessapprox \max_{y \in \mathrm{col}(H) : \|y\| = 1} \|y_\mathrm{m}\|^2$$

where $\|\bullet\|_\mathrm{op}$ denotes the operator norm and $\lessapprox$ denotes inequality up to a proportionality constant. For example, if the i^th column of H is a unit vector, say e_k, then the bound does not guarantee anything. Indeed, if the k^th output is missing, then potentially all information about the i^th latent process is lost. On the other hand, if, for example, $\|U\|_\infty^2 \lessapprox 1/p$, then Corollary 1 in App. O.1 shows that $\varepsilon_\mathrm{rel} \lessapprox s/p$ if s outputs are missing, which means that the approximation will be accurate if $s \ll p$. With this approximation, two things change in the log-likelihood (Rem. 1 in App. O.1): for every time point with missing data *(i)* UU^T becomes $U_\mathrm{o} U_\mathrm{o}^\dagger$ and *(ii)* an extra term $-\frac{1}{2} \log |U_\mathrm{o}^\mathsf{T} U_\mathrm{o}|$ appears.

It is also easy to use variational inference to handle missing data (App. O.2) and to support heterogeneous observation noise (App. P), but we leave experimental tests of these approaches to future work.

3.9. Application to Separable Spatio–Temporal GPs

Separable spatio–temporal GPs, which are of the form $f \sim \mathcal{GP}(0, k_t(t,t')k_r(r,r'))$, form a vector-valued process $f(t) = (f(t,r_i))_{r=1}^p \sim \mathcal{GP}(0, k_t(t,t')K_r)$ when observed at a fixed number of locations in space, 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 OILMM with $H = US^{\frac{1}{2}}$ and $K(t,t') = k_t(t,t')I_p$. Note that $m = p$, so the projection takes $O(np^2)$ time (see App. C).

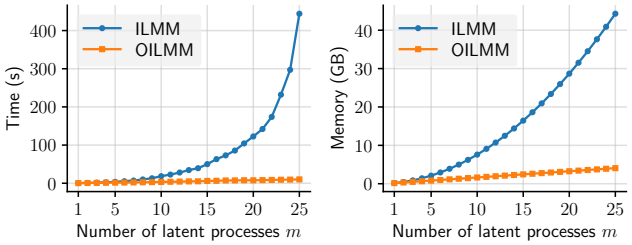


Figure 3. Runtime (left) and memory usage (right) of the ILMM and OILMM for computing the evidence of $n = 1500$ observations for $p = 200$ outputs.

Combining the OILMM framework with efficient state-space scaling techniques (Hartikainen & Särkkä, 2010; Särkkä & Solin, 2019; Solin et al., 2018; Nickisch et al., 2018), which are either exact or arbitrarily good approximations, the complexities are reduced to $O(np^2 + p^3)$ time and $O(np + p^2)$ memory for the entire problem, which are *linear* in n (see App. C). This compares favourably with the filtering techniques of Särkkä et al. (2013) and Hartikainen et al. (2011), both of which have $O(np^3)$ time and $O(np^2)$ memory, and 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.

By relaxing K to be a general diagonal multi-output kernel with $K(t,t) = I_p$, we obtain a new class of models which are nonseparable relaxations of the above in which exact inference remains efficient. The orthogonal basis for this OILMM is, as before, the eigenvectors of a kernel matrix whose hyperparameters can be optimised.

4. Experiments

We test the OILMM in experiments on synthetic and real-world data sets. A Python implementation and code to reproduce the experiments is available at `https://github.com/wesselb/oilmm`. A Julia implementation is available at `https://github.com/willtebbutt/OILMMs.jl`.

4.1. Computational Scaling

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

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

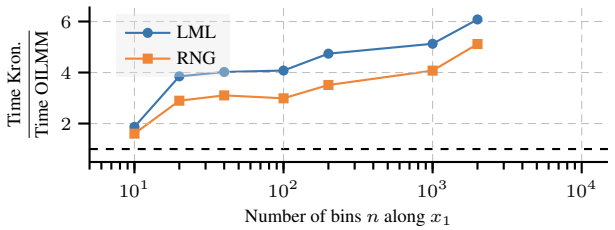


Figure 4. Ratio of timings of the Kronecker approach (Saatçi, 2012, Ch. 5) and the OILMM to compute the marginal likelihood of the latent function (LML) and to generate a single prior sample (RNG). See Tab. 5 in App. R for full results.

Table 1. Root-mean-square error (RMSE) and normalised posterior predictive log-probability (PPLP) of held-out test data for the OILMM with varying m and independent GPs (IGP) in the temperature extrapolation experiment. The OILMM achieves parity in RMSE with IGP at $m = 200$ and surpasses it in PPLP at $m = 5$.

	m	1	5	50	100	200	247
RMSE	OILMM	2.151	2.072	2.030	2.002	1.992	1.991
	IGP	1.993	1.993	1.993	1.993	1.993	1.993
PPLP	OILMM	−1.976	−1.457	−0.905	−0.774	−0.600	−0.525
	IGP	−1.923	−1.923	−1.923	−1.923	−1.923	−1.923

4.2. Rainforest Tree Point Process Modelling

We consider a subset of the extensive rain forest data set credited to Hubbell et al. (2005); Condit (1998); Hubbell et al. (1999) in which the locations of 12929 *trichilia tuberculata* have been recorded. This data is modelled via an inhomogeneous Poisson process, whose log-intensity is given a GP prior. Inference is framed in terms of a latent GP with a Poisson likelihood over a discrete collection of bins. The methodology of Solin et al. (2018) is adapted to accelerate inference of the latent processes, which demonstrates the ability of the OILMM to be combined with existing scaling techniques in a plug-and-play fashion. Inference in the kernel parameters and log-intensity process utilise a simple blocked Gibbs sampler.

It takes roughly three hours[3] to perform 10^5 iterations of MCMC (*circa* 10^5 marginal likelihood evaluations and 10^6 prior samples) with 20000 bins, demonstrating the feasibility of a computationally demanding choice of approximate inference procedure. The Kronecker product factorisation technique (Saatçi, 2012, Ch. 5) is a competitive method in this setting, as it can also efficiently and exactly compute log marginal likelihoods and generate prior samples efficiently. Fig. 4 shows the trade off between the two approaches to inference. In this experiment we define $p = n/2$, meaning that the approach described in Sec. 3.9 scales cubically in n. Despite their quite different implementation details, they do obtain similar performance, with the OILMM performing relatively better as n increases. See App. R for further experimental details and analysis.

4.3. Temperature Extrapolation

Having demonstrated that the OILMM offers computational benefits, we now show that the method can scale to large numbers of latent processes ($m = p = 247$) to capture meaningful dependencies between outputs. 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 OILMM and IGPs (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. For the OILMM, we use a range of numbers of latent processes, up to $m = p = 247$, and let the basis H be 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).

Tab. 1 summarises the performance of the models; more detailed graphs can be found in App. S. The OILMM achieves parity in RMSE with IGP at $m = 200$ latent processes—the data is highly periodic and the predictions are accurate for both models. Moreover, the OILMM requires only $m = 5$ latent processes to achieve a better PPLP than IGP and continues to improve with increasing m, demonstrating the need for a large number of latent processes.

4.4. Exchange Rates Prediction

In this experiment and the next, we test the orthogonality assumption and missing data approximation of the OILMM by comparing its performance to an equivalent ILMM with no restrictions on H and which deals exactly with missing data. We consider daily exchange rates with respect to USD of the top ten international currencies and three precious materials in the year 2007. The task is to predict CAD, JPY, and AUD on particular days given that all other currencies are observed throughout the whole year; we exactly follow Requeima et al. (2019) in the data and setup of the experiment. For the (O)ILMM, we use $m = 3$ latent processes with Matérn–$1/2$ kernels and randomly initialise and learn the basis H.

[3] 3.6 GHz Intel Core i7 processor and 48 GB RAM

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

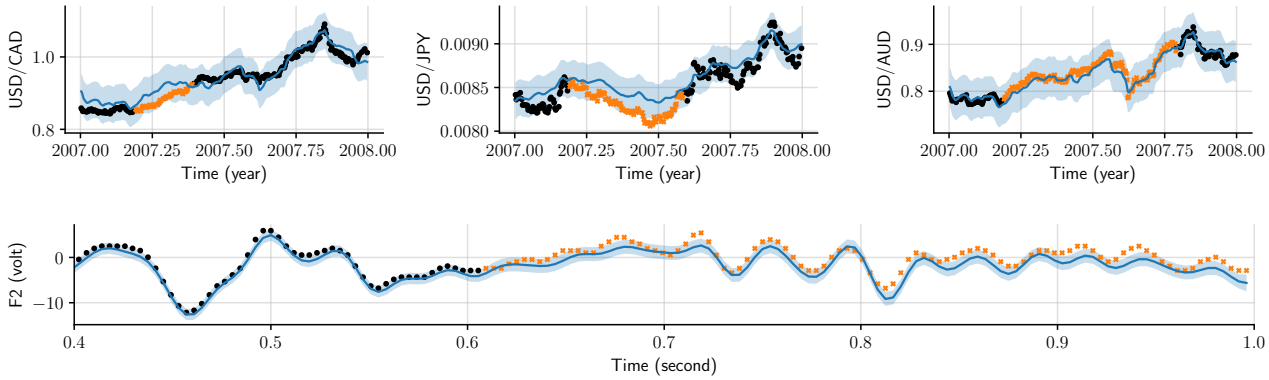


Figure 5. Predictions of the OILMM for the exchange rates experiment (top) and for one of the seven electrodes (F2) in the EEG experiment (bottom). Predictions are shown in blue, denoting the mean and central 95% credible region. Training data are denoted as black dots (•) and held-out test data as orange crosses (×).

Table 2. Standardised mean-squared error (SMSE) and normalised posterior predictive log-probability (PPLP) of held-out test data for various models in the exchange rates (ER) and EEG experiment. IGP stands for independent GPs. The references in square brackets are to models in Fig. 8 in App. B. GPAR (right column) is not a linear MOGP, and thus not comparable to the other methods. However, it is state-of-the-art on both tasks and hence provided as reference. The ILMM and OILMM achieve results equal up to two decimal places. *Numbers are taken from Nguyen & Bonilla (2014). †Numbers are taken from Requeima et al. (2019).

		IGP	CMOGP[11]	CGP[14]	ILMM	OILMM	GPAR[18]
SMSE	ER	0.60*	0.24*	0.21*	0.19	0.19	0.03†
	EEG	1.75†			0.49	0.49	0.26†
PPLP	ER	3.57			3.39	3.39	
	EEG	−1.27			−2.11	−2.11	

Tab. 2 shows that the ILMM and OILMM have identical performance. This shows that the orthogonality assumption and missing data approximation of the OILMM can work well in practice.

4.5. Electroencephalogram Prediction

We consider 256 voltage measurements from 7 electrodes placed on a subject's scalp while the subject is shown a certain image; Zhang et al. (1995) describes the data collection process in detail. The task is to predict the last 100 samples for three electrodes given that the remainder of the data is observed; we exactly follow Requeima et al. (2019) in the data and setup of the experiment. For the (O)ILMM, we use $m = 3$ latent processes with exponentiated quadratic kernels and randomly initialise and learn H.

Tab. 2 shows that the ILMM and OILMM again have identical performance. This again shows that the orthogonality assumption does not harm the model's predictive power and that the missing data approximation can work well.

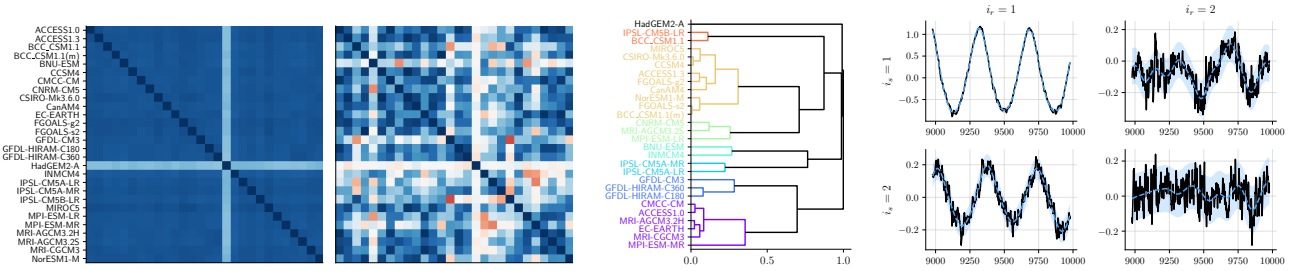
4.6. Large-Scale Climate model Calibration

In this final experiment, we scale to large data in a climate modelling task over Europe. We use the OILMM to find relationships between 28 climate simulators[4] (see Taylor et al., 2012, for background) by letting $H = H_s \otimes H_r$ (see App. H), 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 that the $m_r m_s = 50$ latent processes are indexed by two indices i_s and i_r, one corresponding to the eigenvector of the simulator covariance and one to the eigenvector of the spatial covariance. 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).

Fig. 6a shows that, as opposed to the empirical correlations, which ignore all temporal structure, the correlations learned by the OILMM exhibit a rich structure. A clustering of these correlations in Fig. 6b reveals that the identified structure is meaningful, because structurally similar simulators are grouped near each other. We conclude that the OILMM can be used to analyse large data in a simple and straightforward way, and is able to produce interpretable and meaningful results. See App. T for further experimental details and analysis of the results.

5. Discussion and Conclusion

We investigated the use of a sufficient statistic of the data to accelerate inference in MOGPs with orthogonal bases. In practice, the proposed methodology scales linearly with the number of latent processes m, allowing to scale to large m without sacrificing significant expressivity nor requiring any approximations. This is achieved by breaking down



(a) Empirical (left) and learned (right) correlations (b) Clustering (c) Latent processes

Figure 6. Results of the large-scale climate simulator experiment, showing (a) the empirical correlations and learned correlations (K_s) between the simulators, (b) a dendrogram deriving from hierarchically clustering the simulators based on the learned correlations where the colours indicate discovered groups, 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. Predictions are shown in blue, denoting the mean and central 95% credible region.

the high-dimensional prediction problem into independent single-output problems, whilst retaining exact inference. As a consequence, the method is interpretable, extremely simple to implement, and trivially compatible with off-the-shelf single-output GP scaling techniques for handling large numbers of observations. We tested the method in a variety of experiments, demonstrating that it offers significant computational benefits without harming predictive performance. Interesting future directions are the application to non-Gaussian models for the latent processes (see Prop. 1) and targeting sub-linear time complexity by parallelisation (see, *e.g.*, Särkkä & García-Fernández, 2019).

Acknowledgements

WT acknowledges funding from DeepMind. JSH is supported by EPSRC grant EP/T001569/1, and NERC grant NE/N018028/1. AS acknowledges funding from the Academy of Finland (grant numbers 308640 and 324345). RET is supported by Google, Amazon, ARM, Improbable and EPSRC grants EP/M0269571 and EP/L000776/1.

References

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

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

Álvarez, M., Luengo, D., Titsias, M., and Lawrence, N. D. 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*, pp. 25–32. PMLR, 2010.

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

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

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

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

Bonilla, E. V., Agakov, F. V., and Williams, C. K. I. 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*, pp. 43–50. PMLR, 2007.

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

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

Brochu, E., Cora, V. M., and de Freitas, N. 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.

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

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

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

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

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

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

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

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

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

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

Dee, D. P., Uppala, S., Simmons, A., Berrisford, P., Poli, P., Kobayashi, S., Andrae, U., Balmaseda, M., Balsamo, G., Bauer, d. P., 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.

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

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

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

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

Hartikainen, J. and Särkkä, S. 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)*, pp. 379–384, 2010.

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

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

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

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

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

Higdon, D., Gattiker, J., Williams, B., and Rightley, M. Computer model calibration using high-dimensional output. *Journal of the American Statistical Association*, 103 (482):570–583, 2008. ISSN 01621459. URL http://www.jstor.org/stable/27640080.

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

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

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

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

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

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

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

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

MISO. Historical annual real-time LMPs, 2019. URL https://www.misoenergy.org/markets-and-operations/real-time--market-data/market-reports/#nt=%2FMarketReportType%3AHistorical%20LMP%2FMarketReportName%3AHistorical%20Annual%20Real-Time%20LMPs%20(zip)&t=10&p=0&s=MarketReportPublished&sd=desc.

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

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

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

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

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

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

Osborne, M. A., Roberts, S. J., Rogers, A., Ramchurn, S. D., and Jennings, N. R. 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, pp. 109–120. IEEE Computer Society, 2008.

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

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

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

Roweis, S. and Ghahramani, Z. 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.

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

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

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

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

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

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

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

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

Titsias, M. 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*, pp. 567–574. PMLR, 2009.

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

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

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

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

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

Wu, T., Song, L., Li, W., Wang, Z., Zhang, H., Xin, X., Zhang, Y., Zhang, L., Li, J., Wu, F., 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.

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

Zhang, X., Begleiter, H., Porjesz, B., Wang, W., and Litke, A. Event related potentials during object recognition tasks. *Brain Research Bulletin*, 38(6):531–538, 1995.

Zhe, S., Xing, W., and Kirby, R. M. Scalable high-order Gaussian process regression. In Chaudhuri, K. and Sugiyama, M. (eds.), *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*, pp. 2611–2620. PMLR, 4 2019.

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

Table of Contents

Notation

$\langle \cdot, \cdot \rangle$	Euclidean inner product
$\lVert \cdot \rVert$	Euclidean norm
$\lVert \cdot \rVert_{\text{op}}$	Operator norm
$\lVert \cdot \rVert_{\infty}$	Supremum norm
$\lVert \cdot \rVert_{F}$	Frobenius norm
$S^{\perp}$	Orthogonal complement of S
I_n	$n \times n$ identity matrix
$A > 0$	A is strictly-positive definite
$\lvert A \rvert$	Determinant of A
$A^{\dagger}$	Moore–Penrose pseudo-inverse of A
$\text{col}(A)$	Column space of A
$A \otimes B$	Kronecker product of A and B
$\mathcal{N}(x \mid \mu, \Sigma)$	Density of the multivariate normal distribution with mean μ and covariance Σ at x

Assumptions

Throughout the appendix, we assume that the columns of H are linearly independent and that $\Sigma > 0$. As a consequence, $H^{\mathsf{T}}\Sigma^{-1}H > 0$.

A. How to Implement the OILMM

A.1. Parameters

The parameters of the OILMM are as follows:

Symbol	Type	Description
U	Truncated orthogonal $p \times m$ matrix	Orthogonal part of the basis $H = US^{\frac{1}{2}}$
S	Positive, diagonal $m \times m$ matrix	Diagonal part of the basis $H = US^{\frac{1}{2}}$
σ^2	Positive scalar	Part of the observation noise
D	Positive, diagonal $m \times m$ matrix	Part of the observation noise deriving from the latent processes
$(\theta_i)_{i=1}^m$	Hyperparameters	Hyperparameters for the latent processes, *e.g.* kernel parameters

A.2. Inference

Inference in the OILMM proceeds in three steps. Let $Y \in \mathbb{R}^{p \times n}$ be a matrix where the columns correspond to observations.

Projection step. In the projection step, we project the data to generate "observations for the latent processes". We denote these observations by $Y_{\text{proj}} \in \mathbb{R}^{m \times n}$, where again the columns corresponds to observations. We also construct the "projected noise", which is the observation noise under which the latent processes perform their observations.

(1) Construct the projection:

$$T = S^{-\frac{1}{2}} U^{\mathsf{T}} \in \mathbb{R}^{m \times p}.$$

(2) Project the observations:

$$Y_{\text{proj}} = TY \in \mathbb{R}^{m \times n}.$$

(3) Construct the projected noise:

$$\Sigma_T = \sigma^2 S^{-1} + D \in \mathbb{R}^{m \times m}_{\text{diag}}.$$

This is a diagonal matrix.

The i^{th} row of Y_{proj}, which we denote by $y^{(i)}_{\text{proj}} \in \mathbb{R}^n$, corresponds to observations for latent process i.

Projection step (missing data). In the case of missing data, certain elements of Y are missing. Partition the columns (time stamps) of Y into blocks $Y^{(1)} \in \mathbb{R}^{p \times n_1}, \ldots, Y^{(k)} \in \mathbb{R}^{p \times n_k}$ where $n_1 + \ldots + n_k = n$. These blocks should be chosen such that, for every block $Y^{(i)}$, the observations for an output are either all missing or all available, i.e. every row of $Y^{(i)}$ is either entirely missing or entirely available. Then consider the blocks $Y^{(1)} \in \mathbb{R}^{p \times n_1}, \ldots, Y^{(k)} \in \mathbb{R}^{p \times n_k}$ separately by repeatedly performing inference.

For every block—we henceforth suppress the dependence on the block index—denote by $Y_{\text{o}} \in \mathbb{R}^{p \times n}$ be the rows of the data matrix corresponding to observed outputs. Similarly, let $U_{\text{o}} \in \mathbb{R}^{p \times m}$ be the rows of U corresponding to observed outputs.

(1) Construct the projection:

$$T = S^{-\frac{1}{2}} (U_{\text{o}}^{\mathsf{T}} U_{\text{o}})^{-1} U_{\text{o}}^{\mathsf{T}} \in \mathbb{R}^{m \times p}.$$

(2) Project the observations:

$$Y_{\text{proj}} = TY_{\text{o}} \in \mathbb{R}^{m \times n}.$$

(3) Construct the projected noise:

$$\Sigma_T = \sigma^2 S^{-\frac{1}{2}} d[(U_{\text{o}}{}^{\mathsf{T}} U_{\text{o}})^{-1}] S^{-\frac{1}{2}} + D \in \mathbb{R}^{m \times m}_{\text{diag}}$$

where $d[A]$ sets the off-diagonal elements of A to zero. This is a diagonal matrix.

Latent process inference step. In this step, we perform inference on the latent processes.

(1) For $i = 1, \ldots, m$, do the following:

Conditioning: Condition latent process i on data $y_{\text{proj}}^{(i)} \in \mathbb{R}^n$ where the observation noise is $(\Sigma_T)_{ii}$. The latent process is just an independent GP, and any GP package can be used to do this step. Moreover, any single-output scaling technique can be used here, such as the variational inducing point approximation by Titsias (2009).

Prediction: Make predictions with the posterior of latent process i. Again, any GP package can be used to do this step. Denote the predictive means by $\mu^{(i)} \in \mathbb{R}^n$ and the predictive marginal variances by $\nu^{(i)} \in \mathbb{R}^n$.

(2) Collect the predictive means and marginal variances of the latent processes into matrices μ and ν:

$$\mu = \begin{bmatrix} (\mu^{(1)})^\mathsf{T} \\ \vdots \\ (\mu^{(m)})^\mathsf{T} \end{bmatrix} \in \mathbb{R}^{m \times n}, \quad \nu = \begin{bmatrix} (\nu^{(1)})^\mathsf{T} \\ \vdots \\ (\nu^{(m)})^\mathsf{T} \end{bmatrix} \in \mathbb{R}^{m \times n}.$$

Reconstruction step. In the reconstruction step, we construct the predictions of the OILMM from the predictions of the latent processes.

(1) Construct the basis: $H = US^{\frac{1}{2}} \in \mathbb{R}^{p \times m}$.

(2) Construct the predictive mean of the OILMM:

$$\text{predictive mean} = H\mu \in \mathbb{R}^{p \times n}.$$

(3) Construct the predictive marginal variances of the OILMM:

$$\text{predictive marginal variances} = (H \circ H)\nu \in \mathbb{R}^{p \times n}$$

where $\circ$ denotes the Hadamard product.

A.3. Posterior Sampling

Instead of computing posterior means and marginal variances, you might want to generate posterior samples.

Projection step. See App. A.2.

Latent process sampling step.

(1) For $i = 1, \ldots, m$, do the following:

Conditioning: Condition latent process i on data $y_{\text{proj}}^{(i)} \in \mathbb{R}^n$ where the observation noise is $(\Sigma_T)_{ii}$. The latent process is just an independent GP, and any GP package can be used to do this step. Moreover, any single-output scaling technique can be used here, such as the variational inducing point approximation by Titsias (2009).

Sampling: Sample from the posterior of latent process i. Again, any GP package can be used to do this step. Denote the sample by $\hat{x}^{(i)} \in \mathbb{R}^n$.

(2) Collect the samples into a matrix:

$$\hat{X} = \begin{bmatrix} (x^{(1)})^\mathsf{T} \\ \vdots \\ (x^{(m)})^\mathsf{T} \end{bmatrix} \in \mathbb{R}^{m \times n}.$$

Reconstruction step.

(1) Construct the basis: $H = US^{\frac{1}{2}} \in \mathbb{R}^{p \times m}$.

(2) Construct the posterior sample for the OILMM:

$$\text{posterior sample} = H\hat{X} \in \mathbb{R}^{p \times n}.$$

A.4. Computation of the Log-Marginal Likelihood

Projection step. See App. A.2.

Latent process marginal likelihood calculation.

(1) For $i = 1, \ldots, m$, do the following:

Marginal likelihood: Compute the log-probability of data $y_{\text{proj}}^{(i)} \in \mathbb{R}^n$ under latent process i where the observation noise is $(\Sigma_T)_{ii}$. Denote the resulting log-probability by LML_i. The latent process is just an independent GP, and any GP package can be used to do this step. Moreover, any single-output scaling technique can be used here, such as the variational inducing point approximation by Titsias (2009).

Reconstruction step.

(1) Construct the "regularisation term":

$$\text{regulariser} = -\frac{n}{2}\log|S| - \frac{n(p-m)}{2}\log 2\pi\sigma^2 - \frac{1}{2\sigma^2}(\|Y\|_F^2 - \|U^\mathsf{T} Y\|_F^2)$$

where $\|\bullet\|_F$ denotes the Frobenius norm.

(2) Construct the log-probability of the data Y under the OILMM:

$$\log p(Y) = \text{regulariser} + \sum_{i=1}^{m} \text{LML}_i.$$

Reconstruction step (missing data). In the case of missing data, (1) is slightly different:

(1) Construct the "regularisation term":

$$\text{regulariser} = -\frac{n}{2}\log|S| - \frac{n}{2}\log|U_\text{o}^\mathsf{T} U_\text{o}| - \frac{n(p-m)}{2}\log 2\pi\sigma^2 - \frac{1}{2\sigma^2}(\|Y_\text{o}\|_F^2 - \|\text{chol}(U_\text{o}^\mathsf{T} U_\text{o})^{-1} U_\text{o}^\mathsf{T} Y_\text{o}\|_F^2)$$

where $\|\bullet\|_F$ denotes the Frobenius norm and $\text{chol}(\bullet)$ the Cholesky decomposition. In this case, recall that n corresponds to the number of time points in the current block and to p to the number of observed outputs in the current block.

B. Unifying Presentation of Multi-Output Gaussian Processes

Our attempt at a unifying presentation of MOGP models starts from setting up what we call the *Mixing Model Hierarchy* (MMH). At the bottom of the Mixing Model Hierarchy stands the Instantaneous Linear Mixing Model (ILMM, Mod. 1 in Sec. 2.1), which is a simple, but general class of MOGP models typically characterised by low-rank covariance structure.

The graphical model of the ILMM is illustrated in the top-left corner of Fig. 7, which highlights two restrictions of the ILMM 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 neither H nor $K(t,t) = I_m$ vary with time; and *(ii)* the noise-free observation $f(t)$ is a function of $x(t')$ for $t' = t$ only, meaning that, for example, f cannot be x with a delayed or a smoothed version of x. We hence call the ILMM a *time-invariant* (due to *(i)*) and *instantaneous* (due to *(ii)*) MOGP.

The ILMM 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 mixing mechanism becomes

$$f(t) \,|\, H, x = H(t)x(t).$$

We call such MOGP models *time-varying* (see Fig. 7, 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 mixing mechanism becomes

$$f(t) \,|\, H, x = \int H(t-\tau)x(\tau)\,\mathrm{d}\tau.$$

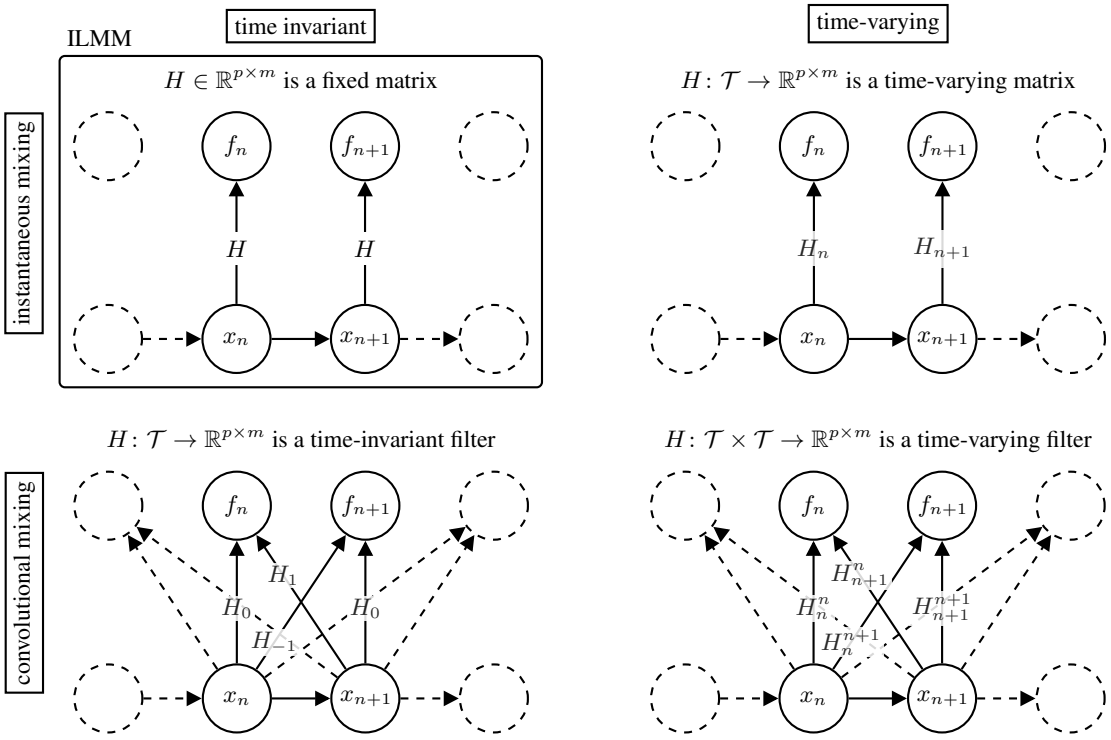
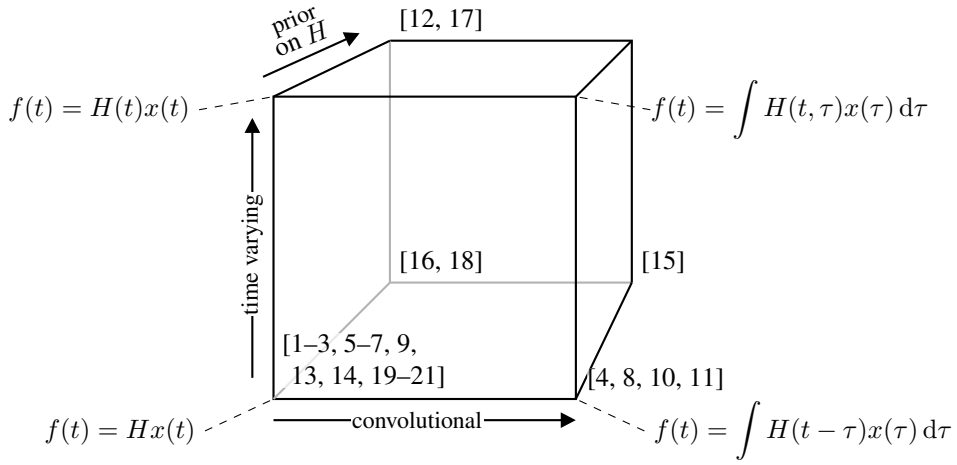


Figure 7. Graphical models illustrating the difference between time-invariant/time-varying and instantaneous/convolutional multi-output GP models, for data sampled at real-valued times $t_1, t_2, \ldots$ (sampling period Δt). Abbreviations used: $x_n = x(t_n)$, $f_n = f(t_n)$, $H_n = H(n\Delta t)$, and $H_n^m = H(t_m, t_n)$. 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 $n' \leq n$.



	Form of H	Form of K	Mixing
[1, 5, 6, 16]	H	$k(t,t')I$	Instantaneous
[2]	$\begin{bmatrix} H_1 & \cdots & H_q \end{bmatrix}$	$\mathrm{diag}(k_1(t,t')I, \ldots, k_q(t,t')I)$	Instantaneous
[3, 7, 9, 13, 20, 21]	H	$\mathrm{diag}(k_1(t,t'), \ldots, k_q(t,t'))$	Instantaneous
[4, 10, 11, 15]	$H(t-t')$	$\mathrm{diag}(\delta(t-t'), \ldots, \delta(t-t'))$	Convolutional
[8]	Green's function	$\mathrm{diag}(k_1(t,t'), \ldots, k_q(t,t'))$	Convolutional
[12, 17]	$H(t)$	$\mathrm{diag}(k_1(t,t'), \ldots, k_q(t,t'))$	Instantaneous
[14]	$\begin{bmatrix} H & I \end{bmatrix}$	$\mathrm{diag}(k_1(t,t'), \ldots, k_{q+p}(t,t'))$	Instantaneous
[18]	Lower triangular	$\mathrm{diag}(k_1(t,t'), \ldots, k_q(t,t'))$	Instantaneous
[19]	$H_1 \otimes \cdots \otimes H_q$	$k(t,t')I$	Instantaneous

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

Figure 8. The Mixing Model Hierarchy, which organises MOGPs from the machine learning and geostatistics literature according to their distinctive modelling assumptions

Table 3. Complexities of learning and inference in the ILMM and OILMM, ignoring the projection. In the table, n is the number of time points; p is the number of outputs; m is the number of latent processes; r is the number of inducing points, typically $r \ll n$; and d is the state dimensionality, typically $d \ll n, m$.

Model	Runtime	Memory
General MOGP	$O(n^3p^3)$	$O(n^2p^2)$
ILMM (Mod. 1)	$O(n^3m^3)$	$O(n^2m^2)$
OILMM (Mod. 2)	$O(n^3m)$	$O(n^2m)$
OILMM (Mod. 2) + Titsias (2009)	$O(nmr^2)$	$O(nmr)$
OILMM (Mod. 2) + Hartikainen & Särkkä (2010)	$O(nmd^3)$	$O(nmd^2)$
APPLICATION TO SEPARABLE SPATIO–TEMPORAL GPS (SEC. 3.9)		
OILMM (Mod. 2)	$O(n^3p)$	$O(n^2p)$
OILMM (Mod. 2) + Titsias (2009)	$O(npr^2)$	$O(npr)$
OILMM (Mod. 2) + Hartikainen & Särkkä (2010)	$O(npd^3)$	$O(npd^2)$
Kronecker product factorisation (Saatçi, 2012, Ch. 5)	$O(n^3+p^3)$	$O(n^2+p^2)$

Table 4. Complexities of projecting the data and reconstructing the predictions in the ILMM and OILMM. In the table, n is the number of time points; p is the number of outputs; and m is the number of latent processes.

Action	Runtime	Memory
Storing data	–	$O(np)$
Construction of projection T	$O(m^2p)$	$O(mp)$
Projection	$O(nmp)$	$O(np)$
Construction of predictive marginal statistics	$O(nmp)$	$O(np)$
APPLICATION TO SEPARABLE SPATIO–TEMPORAL GPS (SEC. 3.9)		
Construction of projection T	$O(p^3)$	$O(p^2)$
Projection	$O(np^2)$	$O(np)$
Construction of predictive marginal statistics	$O(np^2)$	$O(np)$

We call such MOGP models *convolutional* (see Fig. 7, 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 mixing mechanism becomes

$$f(t)\,|\,H, x = \int H(t,\tau)x(\tau)\,\mathrm{d}\tau.$$

We call such MOGP models *time-varying* and *convolutional* (see Fig. 7, bottom right). The graphical models corresponding to these generalisations of the ILMM are depicted in Fig. 7.

The ILMM can be extended in one other way, which is to include a prior distribution on H. This extension and the two previously proposed generalisations together form the *Mixing Model Hierarchy* (MMH), which is depicted in Fig. 8. The MMH organises multi-output Gaussian process models according to their distinctive modelling assumptions. Fig. 8 shows how sixteen MOGP models from the machine learning and geostatistics literature can be recovered as special cases of the various generalisations of the ILMM.

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.

C. Runtime and Memory Complexities

For the ILMM and OILMM, Tab. 3 gives an overview of the runtime and memory complexities associated to learning and inference, and Tab. 4 gives an overview of the runtime and memory complexities associated to projecting the data and reconstructing the predictions.

D. Maximum Likelihood Estimate

Prop. 2. Denote $p(y \mid x) = \mathcal{N}(y \mid Hx, \Sigma)$, and let T be the $m \times p$ matrix $(H^\mathsf{T}\Sigma^{-1}H)^{-1}H^\mathsf{T}\Sigma^{-1}$. Then

$$Ty = \arg\max_x p(y \mid x)$$

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

Proof. Note that

$$\log p(y \mid x) \simeq -\tfrac{1}{2}(y - Hx)^\mathsf{T}\Sigma^{-1}(y - Hx)$$

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

$$x = (H^\mathsf{T}\Sigma^{-1}H)^{-1}H^\mathsf{T}\Sigma^{-1}y = Ty.$$

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

$$\mathbb{E}[Ty \mid x] = THx = (H^\mathsf{T}\Sigma^{-1}H)^{-1}(H^\mathsf{T}\Sigma^{-1}H)x = x. \qquad \square$$

E. Sufficient Statistic

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

Lem. 1.

$$\frac{\mathcal{N}(y \mid Hx, \Sigma)}{\mathcal{N}(y \mid 0, \Sigma)} = \frac{\mathcal{N}(Ty \mid x, (H^\mathsf{T}\Sigma^{-1}H)^{-1})}{\mathcal{N}(Ty \mid 0, (H^\mathsf{T}\Sigma^{-1}H)^{-1})}.$$

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

$$(y - Hx)^\mathsf{T}\Sigma^{-1}(y - Hx) = y^\mathsf{T}\Sigma^{-1}y - 2x^\mathsf{T}H^\mathsf{T}\Sigma^{-1}y + x^\mathsf{T}H^\mathsf{T}\Sigma^{-1}Hx.$$

Here

$$H^\mathsf{T}\Sigma^{-1}y = (H^\mathsf{T}\Sigma^{-1}H)(H^\mathsf{T}\Sigma^{-1}H)^{-1}H^\mathsf{T}\Sigma^{-1}y = (H^\mathsf{T}\Sigma^{-1}H)Ty,$$

so

$$(y - Hx)^\mathsf{T}\Sigma^{-1}(y - Hx) = y^\mathsf{T}\Sigma^{-1}y - 2x^\mathsf{T}(H^\mathsf{T}\Sigma^{-1}H)Ty + x^\mathsf{T}(H^\mathsf{T}\Sigma^{-1}H)x.$$

Adding and subtracting $yT^\mathsf{T}(H^\mathsf{T}\Sigma^{-1}H)Ty$, we find

$$(y - Hx)^\mathsf{T}\Sigma^{-1}(y - Hx) = y^\mathsf{T}\Sigma^{-1}y - yT^\mathsf{T}(H^\mathsf{T}\Sigma^{-1}H)Ty + (x - Ty)^\mathsf{T}(H^\mathsf{T}\Sigma^{-1}H)(x - Ty).$$

Hence, rearranging,

$$(y - Hx)^\mathsf{T}\Sigma^{-1}(y - Hx) - y^\mathsf{T}\Sigma^{-1}y = (x - Ty)^\mathsf{T}(H^\mathsf{T}\Sigma^{-1}H)(x - Ty) - yT^\mathsf{T}(H^\mathsf{T}\Sigma^{-1}H)Ty,$$

which yields the result. $\square$

Prop. 3. The MLE Ty of x is a minimal sufficient statistic for x.

Proof of Prop. 3. By a general characterisation of minimal sufficient statistics (see, *e.g.*, Th. 6.2.13 in Casella & Berger, 2001), Ty is a minimal sufficient statistic for x if and only if it is true that $p(y_1 \mid x)/p(y_2 \mid x)$ is constant as a function of x if and only if $Ty_1 = Ty_2$. Indeed, by Lem. 1,

$$\log \frac{p(y_1 \mid x)}{p(y_2 \mid x)} = (Ty_1 - Ty_2)^\mathsf{T}(H^\mathsf{T}\Sigma^{-1}H)^{-1}x + \text{const.}$$

which, by invertibility of $H^\mathsf{T}\Sigma^{-1}H$, does not depend on x if and only if $Ty_1 = Ty_2$. $\square$

F. Proof of Prop. 1

Proof of Prop. 1. By Prop. 3,

$$p(f \mid Y) = \int p(f \mid x) p(x \mid Y) \, \mathrm{d}x = \int p(f \mid x) p(x \mid TY) \, \mathrm{d}x = p(f \mid TY)$$

where TY are observations for the process Ty. Since

$$y \mid x \sim \mathcal{GP}(Hx, \delta[t - t']\Sigma),$$

the process Ty has distribution

$$Ty \mid x \sim \mathcal{GP}(THx, \delta[t - t']T\Sigma T^\mathsf{T}).$$

By explicit calculation, we find that

$$TH = (H^\mathsf{T}\Sigma^{-1}H)^{-1}H^\mathsf{T}\Sigma^{-1}H = I$$

and

$$T\Sigma T^\mathsf{T} = (H^\mathsf{T}\Sigma^{-1}H)^{-1}H^\mathsf{T}\Sigma^{-1}\Sigma\Sigma^{-1}H(H^\mathsf{T}\Sigma^{-1}H)^{-1} = (H^\mathsf{T}\Sigma^{-1}H)^{-1}.$$

Thus

$$Ty \mid x \sim \mathcal{GP}(x, \delta[t - t']\Sigma_T) \quad \text{where} \quad \Sigma_T = (H^\mathsf{T}\Sigma^{-1}H)^{-1}.$$

Moreover, using Lem. 1, the probability of the data Y is given by

$$p(Y) = \int \prod_{i=1}^{n} \mathcal{N}(y_i \mid Hx, \Sigma) p(x) \, \mathrm{d}x = \left[\frac{\mathcal{N}(y_i \mid 0, \Sigma)}{\mathcal{N}(y_i \mid 0, \Sigma_T)}\right] \int \prod_{i=1}^{n} \mathcal{N}(Ty_i \mid x, \Sigma_T) p(x) \, \mathrm{d}x. \qquad \square$$

G. Interpretation of the Likelihood

Prop. 4. The regularisation terms in like likelihood in Prop. 1 can be written as

$$\log \frac{\mathcal{N}(y \mid 0, \Sigma)}{\mathcal{N}(Ty \mid 0, \Sigma_T)} = -\frac{1}{2}(p - m) \log 2\pi - \overbrace{\frac{1}{2} \log \frac{|\Sigma|}{|\Sigma_T|}}^{\text{noise “lost by projection”}} - \underbrace{\frac{1}{2}\|(I_p - HT)y\|_\Sigma^2}_{\text{data “lost by projection”}},$$

where $\|\bullet\|_\Sigma$ denotes the norm induced by the weighted inner product $\langle \bullet, \bullet \rangle_\Sigma = \langle \Sigma^{-1} \bullet, \bullet \rangle$.

Proof. The first two terms come directly from the multivariate Gaussian densities. We show how the third term may be obtained. Rearrange

$$\langle y, T^\mathsf{T}\Sigma_T^{-1}Ty \rangle = \langle \Sigma^{-\frac{1}{2}}y, (\Sigma^{\frac{1}{2}}T^\mathsf{T}\Sigma_T^{-1}T\Sigma^{\frac{1}{2}})\Sigma^{-\frac{1}{2}}y \rangle = \langle \Sigma^{-\frac{1}{2}}y, P\Sigma^{-\frac{1}{2}}y \rangle$$

where

$$P = \Sigma^{\frac{1}{2}}(T^\mathsf{T}\Sigma_T^{-1}T)\Sigma^{\frac{1}{2}} = \Sigma^{-\frac{1}{2}}H(H^\mathsf{T}\Sigma^{-1}H)^{-1}H^\mathsf{T}\Sigma^{-\frac{1}{2}} = \Sigma^{-\frac{1}{2}}HT\Sigma^{\frac{1}{2}}$$

which is the orthogonal projection onto $\mathrm{col}(\Sigma^{-\frac{1}{2}}H)$. Recall that an orthogonal projection P is defined by $P^2 = P$ and $P^\mathsf{T} = P$. Then

$$\begin{aligned}
\langle y, \Sigma^{-1}y \rangle - \langle y, T^\mathsf{T}\Sigma_T^{-1}Ty \rangle &= \langle \Sigma^{-\frac{1}{2}}y, (I_p - P)\Sigma^{-\frac{1}{2}}y \rangle \\
&= \langle \Sigma^{-\frac{1}{2}}y, (I_p - P)^2\Sigma^{-\frac{1}{2}}y \rangle \\
&= \langle (I_p - P)^\mathsf{T}\Sigma^{-\frac{1}{2}}y, (I_p - P)\Sigma^{-\frac{1}{2}}y \rangle \\
&= \|(I_p - P)\Sigma^{-\frac{1}{2}}y\|^2,
\end{aligned}$$

where we note that $(I_p - P)^2 = I_p - P$ and that $I_p - P$ is symmetric. (In fact, $P^\perp = I_p - P$ is the orthogonal projection onto $\mathrm{col}(\Sigma^{-\frac{1}{2}}H)^\perp$.) To conclude, see that

$$\|(I_p - P)\Sigma^{-\frac{1}{2}}y\|^2 = \|\Sigma^{-\frac{1}{2}}(I_p - \Sigma^{\frac{1}{2}}P\Sigma^{-\frac{1}{2}})y\|^2 = \|(I_p - HT)y\|_\Sigma^2. \qquad \square$$

We note that HT is a projection, but not necessarily an *orthogonal* projection.

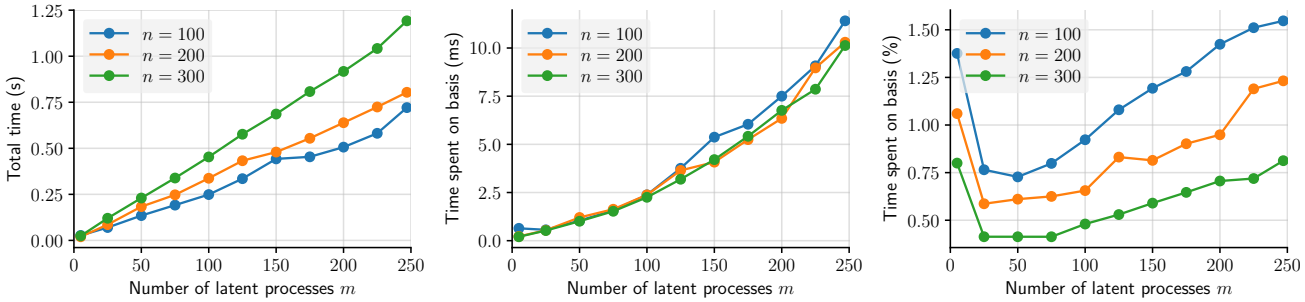


Figure 9. Comparison of the time it takes to construct the basis H to the total time of a log-marginal likelihood computation for a range of numbers of data points n and numbers of latent processes m. The data used is from the temperature extrapolation experiment (Sec. 4.3).

H. Tensor Product Basis

If the observations can be naturally represented as multi-index arrays in $\mathbb{R}^{p_1 \times \cdots \times p_q}$, where the total number of outputs is $p = \prod_{i=1}^{q} p_i$, to obtain a reduction in parameters of H, a natural choice is to correspondingly decompose $H = H_1 \otimes \cdots \otimes H_q$ where $\otimes$ is the Kronecker product and H_i a $p_i \times m_i$ matrix. The latent processes are then naturally seen as a $\mathbb{R}^{m_1 \times \cdots \times m_q}$-valued process, where their total number is $m = \prod_{i=1}^{q} m_i$. In this parametrisation of the ILMM, Prop. 5 shows that the projection and projected noise also become the Kronecker products: $T = T_1 \otimes \cdots \otimes T_q$ and $\Sigma_T = \Sigma_{T_1} \otimes \cdots \otimes \Sigma_{T_q}$. Using the vectorisation trick, TY can be computed efficiently without the need to explicitly construct T.

Prop. 5. Let H be a basis that is a tensor product of other bases and let the observation noise Σ factorise similarly:

$$H = H_1 \otimes \cdots \otimes H_q \quad \text{and} \quad \Sigma = \Sigma_1 \otimes \cdots \otimes \Sigma_q.$$

Then the projection is the tensor product of the projections and the projected noise is the tensor product of the projected noises:

$$T = T_1 \otimes \cdots \otimes T_q \quad \text{and} \quad \Sigma_T = \Sigma_{T_1} \otimes \cdots \otimes \Sigma_{T_q}$$

where $T_i = (H_i^\mathsf{T} \Sigma_i^{-1} H_i)^{-1} H_i^\mathsf{T} \Sigma_i^{-1}$ and $\Sigma_i = (H_i^\mathsf{T} \Sigma_i^{-1} H_i)^{-1}$.

Proof. Follows directly from the compatibility of the Kronecker product with matrix multiplication, transposition, and inversion. □

I. Cost of Parametrising the Basis

For the OILMM, the only computation that does not scale linearly with the number of latent processes m is the parametrisation of the orthogonal part U of the basis H, which takes $O(m^2 p)$ time. We argue that this cost is dominated by the cost $O(n^3 m + nmp)$ of computing the log-marginal likelihood of the projected data:

(i) typically $m \leq n, p$;

(ii) the cost of computing the log-marginal likelihood of the projected data scales with n, and often $n \gg m, p$; and

(iii) assuming that p is not much bigger than n, computing the log-marginal likelihood of the projected data costs at least $O(n)$ more, so the cost of parametrising the basis H should become insignificant as n grows.

We compare the time it takes to construct the basis H to the total time of a log-marginal likelihood computation for a range of numbers of data points n and numbers of latent processes m. We use the data from the temperature extrapolation experiment (Sec. 4.3). The results are depicted in Fig. 9. Observe that, even in the worst case when $m = p = 247$, parametrising the basis H takes no more than 1.5% of the total time at $n = 100$ data points and no more than 0.8% of the total time at $n = 300$ data points. This cost is negligible, even in this worst case.

J. Characterisation of Diagonal Projected Noise

Prop. 6 says that the projected noise is diagonal if and only if H is of the form $H = \Sigma^{\frac{1}{2}} U S^{\frac{1}{2}}$ with U a matrix with orthonormal columns and $S > 0$ diagonal. This condition is awkward, as it couples H and Σ. Fortunately, Prop. 6 also

shows that we may drop H's dependency on Σ if and only if every column of U is an eigenvector of Σ.

Prop. 6. The projected noise Σ_T is diagonal if and only if H is of the form $H = \Sigma^{\frac{1}{2}} U S^{\frac{1}{2}}$ with U a matrix with orthonormal columns 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 Σ.

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

$$S^{-\frac{1}{2}} H^\mathsf{T} \Sigma^{-\frac{1}{2}} \Sigma^{-\frac{1}{2}} H S^{-\frac{1}{2}} = I_m,$$

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

For the second statement, note that every column of U is an eigenvector of Σ if and only if it is an eigenvector of $\Sigma^{\frac{1}{2}}$. Suppose that H is of the form $H = UD^{\frac{1}{2}}$ with $D > 0$ diagonal. Then

$$\Sigma^{\frac{1}{2}} U = \Sigma^{\frac{1}{2}} U S^{\frac{1}{2}} S^{-\frac{1}{2}} = H S^{-\frac{1}{2}} = U D^{\frac{1}{2}} S^{-\frac{1}{2}},$$

so every column of U is an eigenvector of $\Sigma^{\frac{1}{2}}$. Conversely, suppose that every column of U is an eigenvector of $\Sigma^{\frac{1}{2}}$ with eigenvalues stacked into a diagonal matrix $D > 0$. Then

$$H = \Sigma^{\frac{1}{2}} U S^{\frac{1}{2}} = U D S^{\frac{1}{2}},$$

which is of the desired form. □

K. Kullback–Leibler Divergence Between an ILMM and OILMM

Prop. 7. Consider two ILMMs with equal $\Sigma = \sigma^2 I_p$, 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

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

and

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

where $\hat{H}$ ranges over matrices of the form $US^{\frac{1}{2}}$ with U a matrix with orthonormal columns 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 ILMM before adding noise.

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

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

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

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

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 $H = USV^\mathsf{T}$ 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 $U^\mathsf{T}U = I_m$, but $UU^\mathsf{T} \neq I_p$. Then, choosing $\hat{H} = US$,

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

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

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

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

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

since

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

Therefore,

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

where the equality follows from a similar calculation:

$$\|I_m - V\|_F^2 = \operatorname{tr}[I_m - V^\mathsf{T} - V + V^\mathsf{T}V] = 2\operatorname{tr}[I_m - V].$$ □

L. OILMM: Projection and Projected Noise

Prop. 8. Consider the OILMM (Mod. 2). Then the projection and projected noise are given by

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

Proof. To begin with, note that

$$\begin{aligned} y &\sim \mathcal{GP}(\underbrace{HK(t,t')H^\mathsf{T} + \delta[t - t'](\sigma^2 I_p + HDH^\mathsf{T})}), \\ &= H(K(t,t') + \delta[t - t']D)H^\mathsf{T} + \delta[t - t']\sigma^2 I_p, \end{aligned}$$

so we can assume that $D = 0$ by "absorbing it into $K(t,t')$". We then find that

$$H^\mathsf{T}\Sigma^{-1}H = \sigma^{-2}S,$$

so

$$\Sigma_T = T\Sigma T^\mathsf{T} = (H^\mathsf{T}\Sigma^{-1}H)^{-1} = \sigma^2 S^{-1}.$$

Moreover, then

$$T = (H^\mathsf{T}\Sigma^{-1}H)^{-1}H^\mathsf{T}\Sigma^{-1} = (\sigma^2 S^{-1})(\sigma^{-2}S^{\frac{1}{2}}U^\mathsf{T}) = S^{-\frac{1}{2}}U^\mathsf{T}.$$

Finally, "pull D back out of $K(t,t')$", which we note is equivalent to adding it to Σ_T by Prop. 1. □

M. OILMM: Likelihood

Prop. 9. Consider the OILMM (Mod. 2). Let Y be an $p \times n$ matrix of observations for y. Then

$$\log p(Y) = -\frac{n}{2}\log|S| - \frac{n(p-m)}{2}\log 2\pi\sigma^2 - \frac{1}{2\sigma^2}\|(I_p - UU^\mathsf{T})Y\|_F^2 + \sum_{i=1}^{m}\log\mathcal{N}((TY)_{i:} \mid 0, K_i + (\sigma^2/S_{ii} + D_{ii})I_n)$$

where $\|\cdot\|_F$ denotes the Frobenius norm and K_i is the $n \times n$ kernel matrix for the i^{th} latent process x_i.

Proof. By Prop. 1 and Prop. 4, we have

$$\log p(Y) = -\frac{n(p-m)}{2}\log 2\pi - \frac{n}{2}\log\frac{|\Sigma|}{|\Sigma_T|} - \frac{1}{2}\sum_{i=1}^{n}\|(I_p - HT)y_i\|_\Sigma^2 + \log\int p(x)\prod_{i=1}^{n}\mathcal{N}(Ty_i \mid x_i, \Sigma_T)\,\mathrm{d}x.$$

Using the same trick as in the proof of Prop. 8, assume that $D = 0$ by "absorbing it into $K(t,t')$". We then simplify the terms one by one. First, we have that

$$\log\frac{|\Sigma|}{|\Sigma_T|} = \log\frac{|\sigma^2 I_p|}{|\sigma^2 S^{-1}|} = (p-m)\log\sigma^2 + \log|S|.$$

Second, note that $I_p - HT = I_p - UU^\mathsf{T}$, which we denote by $P_{U^\perp}$ and which is symmetric, so

$$\|(I_p - HT)y_i\|_\Sigma^2 = \|P_{U^\perp}y_i\|_\Sigma^2 = \langle P_{U^\perp}y_i, \Sigma^{-1}P_{U^\perp}y_i\rangle = \sigma^{-2}\langle P_{U^\perp}y_i, P_{U^\perp}y_i\rangle = \sigma^{-2}\operatorname{tr}[P_{U^\perp}P_{U^\perp}y_i y_i^\mathsf{T}].$$

Then sum over $i = 1, \ldots, n$ to obtain

$$\sum_{i=1}^{n}\|(I_p - HT)y_i\|_\Sigma^2 = \sigma^{-2}\operatorname{tr}[P_{U^\perp}P_{U^\perp}YY^\mathsf{T}] = \sigma^{-2}\|P_{U^\perp}Y\|_F^2.$$

Finally,

$$\log\int p(x)\prod_{i=1}^{n}\mathcal{N}(Ty_i \mid x_i, \Sigma_T)\,\mathrm{d}x = \sum_{i=1}^{m}\log\mathcal{N}((TY)_{i:} \mid 0, K_i + (\sigma^2/S_{ii} + D_{ii})I_n)$$

follows from independence of the latent processes and remembering that we "absorbed D into $K(t,t')$". □

Observe that

$$\|(I_p - UU^\mathsf{T})Y\|_F^2 = \|Y\|_F^2 - \|U^\mathsf{T}Y\|_F^2,$$

which is a computationally more efficient implementation.

N. OILMM: Decomposition of the Mean Squared Error

Prop. 10. Let $H = US^{\frac{1}{2}}$ with U a matrix with orthonormal columns and $S^{\frac{1}{2}} > 0$ diagonal. Then

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

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

Proof. By expanding and using orthogonality of U,

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

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

$$\langle y, (I_p - UU^\mathsf{T})y\rangle = \langle y, P_{U^\perp}y\rangle = \langle y, P_{U^\perp}^2 y\rangle = \langle P_{U^\perp}^\mathsf{T}y, P_{U^\perp}y\rangle = \langle P_{U^\perp}y, P_{U^\perp}y\rangle = \|P_{U^\perp}y\|^2. \qquad \square$$

O. OILMM: Missing Data

For a matrix or vector A, let A_o and A_m denote the rows of A corresponding to respectively observed and missing values.

Prop. 11. Consider the OILMM (Mod. 2). For observed outputs y_o, which are a subset of all outputs y, the projection and projected noise are given by

$$T_\mathrm{o} = S^{-\frac{1}{2}}U_\mathrm{o}^\dagger \quad \text{and} \quad \Sigma_{T_\mathrm{o}} = \sigma^2 S^{-\frac{1}{2}}(U_\mathrm{o}^\mathsf{T}U_\mathrm{o})^{-1}S^{-\frac{1}{2}} + D$$

where $U_\mathrm{o}^\dagger$ is the pseudo-inverse of U_o.

Proof. Note that

$$y_\mathrm{o} \sim \mathcal{GP}(H_\mathrm{o}K(t,t')H_\mathrm{o}^\mathsf{T} + \delta[t-t'](\sigma^2 I_\mathrm{o} + H_\mathrm{o}DH_\mathrm{o}^\mathsf{T})),$$

so y_o is an ILMM with basis H_o and observation noise $\sigma^2 I + H_\mathrm{o}DH_\mathrm{o}^\mathsf{T}$. The proof proceeds like that of Prop. 8, also using trick of assuming that $D = 0$ by "absorbing it into $K(t,t')$". To begin with, we have

$$H^\mathsf{T}\Sigma^{-1}H = \sigma^{-2}S^{\frac{1}{2}}U_\mathrm{o}^\mathsf{T}U_\mathrm{o}S^{\frac{1}{2}},$$

so

$$\Sigma_T = T\Sigma T^\mathsf{T} = (H^\mathsf{T}\Sigma^{-1}H)^{-1} = \sigma^2 S^{-\frac{1}{2}}(U_\mathrm{o}^\mathsf{T}U_\mathrm{o})^{-1}S^{-\frac{1}{2}}.$$

Moreover, then

$$T = (H^\mathsf{T}\Sigma^{-1}H)^{-1}H^\mathsf{T}\Sigma^{-1} = (\sigma^2 S^{-\frac{1}{2}}(U_\mathrm{o}^\mathsf{T}U_\mathrm{o})^{-1}S^{-\frac{1}{2}})(\sigma^{-2}S^{\frac{1}{2}}U_\mathrm{o}^\mathsf{T}) = S^{-\frac{1}{2}}(U_\mathrm{o}^\mathsf{T}U_\mathrm{o})^{-1}U_\mathrm{o}^\mathsf{T} = S^{-\frac{1}{2}}U_\mathrm{o}^\dagger.$$

Finally, "pull D back out of $K(t,t')$", which, again, is equivalent to adding it to Σ_T. $\square$

Rem. 1. When using T_o and Σ_{T_o}, in the likelihood computation in Prop. 9, from Prop. 4, it can be seen that two things change: for every time point with missing data,

(1) $H_\mathrm{o}T_\mathrm{o} = U_\mathrm{o}U_\mathrm{o}^\dagger$, so UU^T becomes $U_\mathrm{o}U_\mathrm{o}^\dagger$; and

(2) $\frac{1}{2}\log|\Sigma_{T_\mathrm{o}}|$ gives an extra term $-\frac{1}{2}\log|U_\mathrm{o}^\mathsf{T}U_\mathrm{o}|$.

O.1. Diagonal Approximation of Projected Noise

For a matrix A, let $d[A]$ denote the diagonal matrix resulting from setting the off-diagonal entries of A to zero.

Prop. 12. For Σ_{T_o} from Prop. 11, we have

$$\frac{\|\Sigma_{T_\mathrm{o}} - d[\Sigma_{T_\mathrm{o}}]\|_{\mathrm{op}}}{\|d[\Sigma_{T_\mathrm{o}}]\|_{\mathrm{op}}} \leq \frac{S_{\max}}{S_{\min}} \max_{y \in \mathrm{col}(H):\|y\|=1} \|y_\mathrm{m}\|^2$$

where $\|\bullet\|_{\mathrm{op}}$ denotes the operator norm, and $S_{\min}$ and $S_{\max}$ are the smallest and largest diagonal values of S.

Proof. Let e_i be the i^{th} unit vector. Denote $A = (U_\mathrm{o}^\mathsf{T} U_\mathrm{o})^{-1}$, and let $\lambda_{\min}$ and $\lambda_{\max}$ be the minimum and maximum eigenvalue of A. To begin with,

$$d[A]_{ii} = \langle e_i, Ae_i \rangle \in [\lambda_{\min}, \lambda_{\max}].$$

Let $x \in \mathbb{R}^m$ be such that $\|x\| = 1$. Then

$$\langle x, (A - d[A])x \rangle = \langle x, Ax \rangle - \langle x, d[A]x \rangle \leq \lambda_{\max} - \lambda_{\min}.$$

Similarly,

$$\langle x, (A - d[A])x \rangle \geq -\lambda_{\max} + \lambda_{\min}.$$

Therefore,

$$|\langle x, (A - d[A])x \rangle| \leq \lambda_{\max} - \lambda_{\min},$$

so

$$\|A - d[A]\|_{\mathrm{op}} \leq \lambda_{\max} - \lambda_{\min}$$

Since S is diagonal, we have

$$\Sigma_{T_\mathrm{o}} - d[\Sigma_{T_\mathrm{o}}] = \sigma^2 S^{-\frac{1}{2}} (A - d[A]) S^{-\frac{1}{2}}.$$

Using the derived bound on the operator norm and submultiplicativity of the operator norm, it follows that

$$\|\Sigma_{T_\mathrm{o}} - d[\Sigma_{T_\mathrm{o}}]\|_{\mathrm{op}} \leq \sigma^2 S_{\min}^{-1} (\lambda_{\max} - \lambda_{\min}).$$

Moreover,

$$\|d[\Sigma_{T_\mathrm{o}}]\|_{\mathrm{op}} = \sigma^2 \max_{i=1,\ldots,m} (S_{ii}^{-1} d[A]_{ii} + D_{ii}) \geq \sigma^2 \max_{i=1,\ldots,m} S_{ii}^{-1} d[A]_{ii} \geq \sigma^2 S_{\max}^{-1} \max_{i=1,\ldots,m} d[A]_{ii} \geq \sigma^2 S_{\max}^{-1} \lambda_{\max}.$$

Therefore,

$$\frac{\|\Sigma_{T_\mathrm{o}} - d[\Sigma_{T_\mathrm{o}}]\|_{\mathrm{op}}}{\|d[\Sigma_{T_\mathrm{o}}]\|_{\mathrm{op}}} \leq \frac{S_{\max}}{S_{\min}} \left(1 - \frac{\lambda_{\min}}{\lambda_{\max}}\right).$$

By definition of $\lambda_{\min}$ and $\lambda_{\max}$ and orthogonality of U, we have that

$$\frac{1}{\lambda_{\min}} = \max_{x \in \mathbb{R}^m : \|x\|=1} \|U_\mathrm{o} x\|^2 \leq \max_{x \in \mathbb{R}^m : \|x\|=1} \|Ux\|^2 = 1 \quad \text{and} \quad \frac{1}{\lambda_{\max}} = \min_{x \in \mathbb{R}^m : \|x\|=1} \|U_\mathrm{o} x\|^2.$$

Substitute these results into the bound:

$$\frac{\|\Sigma_{T_\mathrm{o}} - d[\Sigma_{T_\mathrm{o}}]\|_{\mathrm{op}}}{\|d[\Sigma_{T_\mathrm{o}}]\|_{\mathrm{op}}} \leq \frac{S_{\max}}{S_{\min}} \left(1 - \min_{x \in \mathbb{R}^m : \|x\|=1} \|U_\mathrm{o} x\|^2\right) = \frac{S_{\max}}{S_{\min}} \max_{x \in \mathbb{R}^m : \|x\|=1} (1 - \|U_\mathrm{o} x\|^2).$$

By orthogonality of U, for $x \in \mathbb{R}^m$ such that $\|x\| = 1$, we have

$$1 = \|x\|^2 = \|Ux\|^2 = \|U_\mathrm{o} x\|^2 + \|U_\mathrm{m} x\|^2,$$

so $1 - \|U_\mathrm{o} x\|^2 = \|U_\mathrm{m} x\|^2$. Therefore,

$$\max_{x \in \mathbb{R}^m : \|x\|=1} (1 - \|U_\mathrm{o} x\|^2) = \max_{x \in \mathbb{R}^m : \|x\|=1} \|U_\mathrm{m} x\|^2 = \max_{x \in \mathbb{R}^m : \|x\|=1} \|(Ux)_\mathrm{m}\|^2 = \max_{y \in \mathrm{col}(U) : \|y\|=1} \|y_\mathrm{m}\|^2$$

and we conclude by noting that $\mathrm{col}(U) = \mathrm{col}(H)$. □

Cor. 1. Suppose $\|U\|_\infty^2 \leq C/p$ for some $C \geq 1$, and that s outputs are missing. Then

$$\frac{\|\Sigma_{T_\text{o}} - d[\Sigma_{T_\text{o}}]\|_\text{op}}{\|d[\Sigma_{T_\text{o}}]\|_\text{op}} \leq C \frac{S_\text{max}}{S_\text{min}} \frac{ms}{p}.$$

Proof. Let $y \in \text{col}(H)$ be such that $\|y\| = 1$. Then $y = Ux$ for some $x \in \mathbb{R}^m$ such that $\|x\| = 1$. Therefore,

$$\|y_\text{m}\|^2 = \sum_{i \in \text{missing}} (Ux)_i^2 \leq \sum_{i \in \text{missing}} \|U_{i:}\|^2 \|x\|^2 = \sum_{i \in \text{missing}} \|U_{i:}\|^2 \leq \frac{Cms}{p},$$

so the result follows from the previous proposition. □

O.2. Variational Approach

Let Y_o be the observed data. Complement Y_o with missing data Y_m such that $Y = Y_\text{o} \cup Y_\text{m}$ is complete. Then a way to deal with missing data is to use variational inference. In particular, assume a Gaussian approximate posterior distribution $q(Y_\text{m})$ over Y_m, and maximise the evidence lower bound (ELBO) $\mathcal{L}$ using gradient-based optimisation:

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

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

P. OILMM: Heterogeneous Observation Noise

Although the specification of the observation noise $\Sigma = \sigma^2 I_p + HDH^\mathsf{T}$ in the OILMM does not allow for heterogeneous observation noise, it is possible to set $\Sigma = \text{diag}(\sigma_1^2, \ldots, \sigma_p^2)$ and use Prop. 6 to include Σ in the parametrisation of H: $H = \Sigma^{\frac{1}{2}} U S^{\frac{1}{2}}$. 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 $\Sigma^{-\frac{1}{2}}$. 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: $\langle h_i, h_j \rangle_\Sigma = \sum_{k=1}^p h_{ik} h_{jk} / \sigma_k^2 = 0$ 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 Σ and H.

Q. Computational Scaling Experiment (Sec. 4.1) Additional Details

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.

R. Point Process Experiment (Sec. 4.2) Additional Details and Analysis

We consider a subset of the extensive rainforest data set credited to Hubbell et al. (2005); Condit (1998); Hubbell 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. 10).

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 $\Sigma(x)$ is modelled with a Gaussian process such that $f(x) = \log \Sigma(x)$. Locally-constant intensity in subregions are modelled by discretising the region into np bins (Møller et al., 1998). This leads to a Poisson observation model for each bin. This model reaches posterior consistency in the limit of bin width going

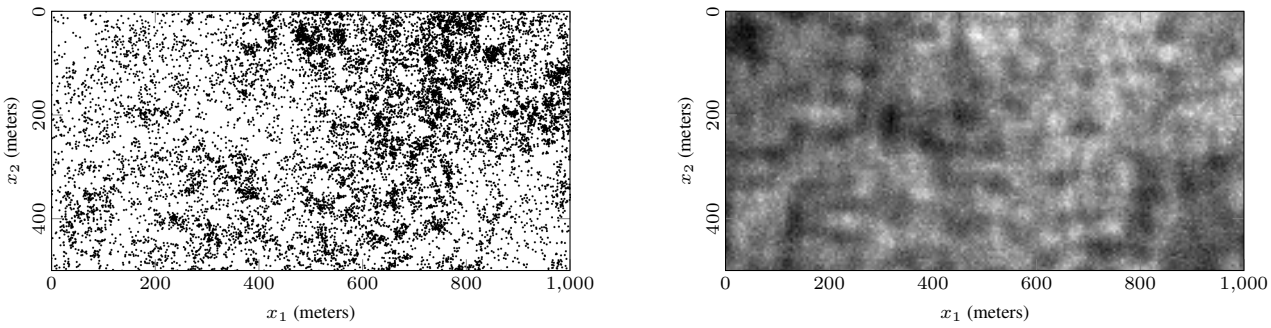
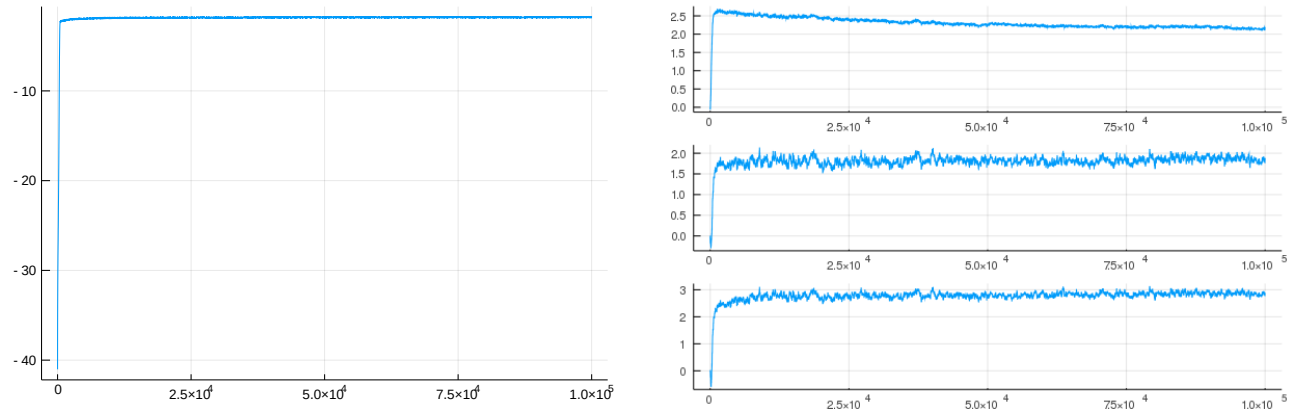


Figure 10. Observations of the rainforest tree locations (left), and posterior mean log-intensity for the log-Gaussian Cox process model (right) with a grid of $np = 20000$ observation bins.



(a) (b)

Figure 11. (a): Log–joint probability per iteration. (b): Hyperparameters per iteration. Shows the length scale, process variance, and nugget variance respectively.

to zero (Tokdar & 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 $n \times p = 200 \times 100$ (each bin is $5\,\mathrm{m} \times 5\,\mathrm{m}$) grid with $np = 20000$ grid bins in total, and treat the first dimension as time. The conditional probability of the complete binned data set given the latent GP is therefore

$$p(Y \mid f) \approx \prod_{i=1}^{n} \prod_{j=1}^{p} \mathrm{Poisson}(Y_{ij} \mid a e^{f(r_{ij})}),$$

where r_{ij} is the coordinate of the ij^{th} bin, Y_{ij} is the number of data points in the ij^{th} bin, Y is the $n \times p$ matrix of counts, and a is the area of each bin.

We perform 10^5 iterations of block Gibbs sampling, each of which comprises 10 iterations Elliptical Slice Sampling (Murray et al., 2010; Murray & Adams, 2010) for the Gaussian process given its hyperparameters, and a single iteration of Metropolis Hastings (Hastings, 1970) with proposal distribution $\mathcal{N}(\theta, 0.05^2)$ for the log of the hyperparameters given the latent GP-distributed function. Each step of Elliptical Slice Sampling requires an additional sample from the GP prior at the current hyperparameter values, while each step of Metropolis Hastings requires a log marginal likelihood evaluation. As such approximately 10^6 samples from the prior were drawn, and 10^5 log marginal likelihood calculations undertaken. The kernel is a product of two Matérn-$5/2$ kernels with a shared length scale. A single process variance is utilised, and a nugget term is added. The log of each of the three hyperparameters was given a $\mathcal{N}(0, 1)$ prior. Fig. 11a shows the log joint of the entire state after each iteration, while Fig. 11b shows the progress of each hyperparameter per iteration.

The times in Fig. 4 were obtained via `BenchmarkTools.jl` (Chen & 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

Table 5. Description of the data points associated with the timing experiment from Fig. 4

n	LML Kronecker	OILMM	RNG Kronecker	OILMM
2000	2.45 ± 0.0193	0.403 ± 0.00414	2.45 ± 0.0278	0.478 ± 0.00376
1000	0.365 ± 0.00256	0.0712 ± 0.000369	0.364 ± 0.00451	0.0892 ± 0.000435
200	0.0111 ± 0.000301	$0.00235 \pm 2.53 \times 10^{-5}$	$0.0112 \pm 9.89 \times 10^{-5}$	$0.00318 \pm 1.2 \times 10^{-5}$
100	$0.00237 \pm 8.66 \times 10^{-6}$	$0.000582 \pm 6.55 \times 10^{-7}$	$0.00237 \pm 3.1 \times 10^{-5}$	$0.000792 \pm 8.69 \times 10^{-7}$
40	$0.00044 \pm 4.35 \times 10^{-7}$	$0.000109 \pm 2.22 \times 10^{-7}$	$0.000436 \pm 3.19 \times 10^{-7}$	$0.000141 \pm 2.0 \times 10^{-7}$
20	$9.15 \times 10^{-5} \pm 1.48 \times 10^{-7}$	$2.38 \times 10^{-5} \pm 2.1 \times 10^{-7}$	$9.06 \times 10^{-5} \pm 1.89 \times 10^{-7}$	$3.13 \times 10^{-5} \pm 1.72 \times 10^{-7}$
10	$1.84 \times 10^{-5} \pm 1.54 \times 10^{-7}$	$9.87 \times 10^{-6} \pm 1.08 \times 10^{-7}$	$1.84 \times 10^{-5} \pm 3.02 \times 10^{-7}$	$1.15 \times 10^{-5} \pm 1.17 \times 10^{-7}$

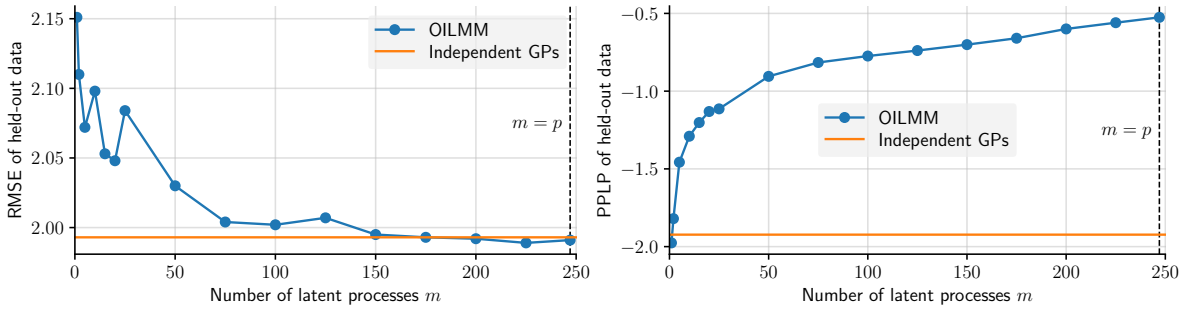


Figure 12. RMSE and PPLP achieved in the temperature extrapolation experiment.

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.

When computing the log marginal likelihood, the state-space implementation of the GP makes use of the infinite-horizon trick introduced to the GP literature by Solin et al. (2018). However, this trick is only exploited here 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}. This produces log marginal likelihood evaluations and samples from the prior that are exact for all practical purposes.

R.1. Performance versus Kronecker Trick

Fig. 4 demonstrates that, for the particular approach taken to inference in the Poisson process and, importantly, the dimensions of the data, the Kronecker trick discussed by Saatçi (2012) takes slightly longer to compute log marginal likelihoods and generate samples than does the OILMM implemented in the manner described above. It would of course be unreasonable to assert that the OILMM dominates the Kronecker trick; rather, it seems appropriate to assert that they are competitive with each other in the regime considered.

This is perhaps surprising as the performance of the Kronecker trick is determined almost entirely by a couple of computationally intensive operations, the eigendecomposition and matrix-matrix multiplies. Carefully optimised implementations of these operations exist, and were used, to implement the Kronecker trick. Conversely, the OILMM implementation discussed above comprises many small operations. While our implementation benefits from *e.g.* the `StaticArrays.jl` library, which is suitable for operations on small matrices and vectors, it remains surprising that similar performance was found.

In general we anticipate the OILMM implemented in the described manner be significantly faster on data sets where n is much larger than p, whilst the Kronecker trick will likely do better when n is similar to p.

S. Temperature Extrapolation Experiment (Sec. 4.3) Additional Results

Fig. 12 depicts the RMSE and PPLP achieved in the temperature extrapolation experiment (Sec. 4.3).

T. Large-Scale Climate Model Calibration Experiment (Sec. 4.6) Additional Details and Analysis

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 & Wright, 2006), which takes about 4 hours on a MacBook Pro (2.7 GHz Intel Core i7 processor and 16 GB RAM). The learned length scales were $23.3°$ for latitude and $43.6°$ for longitude.

Fig. 6a shows the empirical correlations and the correlations learned by the OILMM (derived from K_s). 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. 6b 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. 6a and 6b, 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. 6b, 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.[5]

(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. 6c 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.

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