
Supplementary Material for Multi-resolution Multi-task Gaussian Processes

Oliver Hamelijnck
The Alan Turing Institute
Department of Computer Science
University of Warwick
ohamelijnck@turing.ac.uk

Theodoros Damoulas
The Alan Turing Institute
Depts. of Computer Science & Statistics
University of Warwick
tdamoulas@turing.ac.uk

Kangrui Wang
The Alan Turing Institute
Department of Statistics
University of Warwick
kwang@turing.ac.uk

Mark A. Girolami
The Alan Turing Institute
Department of Engineering
University of Cambridge
mgirolami@turing.ac.uk

Abstract

We consider evidence integration from potentially dependent observation processes under varying spatio-temporal sampling resolutions and noise levels. We offer a multi-resolution multi-task (MRGP) framework that allows for both *inter-task* and *intra-task* multi-resolution and multi-fidelity. We develop shallow Gaussian Process (GP) mixtures that approximate the difficult to estimate joint likelihood with a composite one and deep GP constructions that learn mappings between resolutions and naturally handle biases. In doing so, we generalize existing approaches and offer information-theoretic corrections and efficient variational approximations. We demonstrate the competitiveness of MRGPs on synthetic settings and on the challenging problem of hyper-local estimation of air pollution levels across London from multiple sensing modalities operating at disparate spatio-temporal resolutions.

1 Introduction

The increased availability of ground and remote sensing networks coupled with new modalities, arising from e.g. citizen science initiatives and mobile platforms, is creating new challenges for performing formal evidence integration. These multiple observation processes and sensing modalities can be dependent, with different signal-to-noise ratios and varying sampling resolutions across space and time. In our motivating application, London authorities measure air pollution from multiple sensor networks; high-fidelity ground sensors that provide frequent multi-pollutant readings, low fidelity diffusion tubes that only provide monthly single-pollutant readings, hourly satellite-derived information at large spatial scales, and high frequency medium-fidelity multi-pollutant sensor networks. Such a multi-sensor multi-resolution multi-task evidence integration setting is becoming prevalent across many real world applications of spatio-temporal problems.

The current state of the art, see also Section 5, is assuming independent and unbiased observation processes and cannot handle the challenges of real world settings that are jointly *non-stationary*, *multi-task*, *multi-fidelity*, and *multi-resolution* [2, 7, 15, 23, 24, 29, 30]. The latter challenge has recently attracted the interest of the machine learning community under the context of working with aggregate, binned observations [2, 15, 30] or the special case of natural language generation at multiple levels of abstraction [29]. When the independence and unbiasedness assumptions are not

satisfied they lead to posterior contraction, degradation of predictive performance and insufficient uncertainty quantification.

In this paper we introduce a multi-resolution multi-task GP framework that can integrate evidence from observation processes with varying support (e.g. partially overlapping in time and space), that can be dependent and biased while allowing for both *inter-task* and *intra-task* multi-resolution and multi-fidelity. Our first contribution is a shallow GP mixture, MR-GPRN, that corrects for the dependency between observation processes through composite likelihoods and extends the Gaussian aggregation model of Law et al. [15], the multi-task GP model of Wilson et al. [36], and the variational lower bound of Nguyen and Bonilla [20]. Our second contribution is a multi-resolution deep GP composition that can additionally handle biases in the observation processes and extends the deep GP models and variational lower bounds of Damianou and Lawrence [5] and Salimbeni and Deisenroth [28] to varying support, multi-resolution data. Lastly, we demonstrate the superiority of our models on synthetic problems and on the challenging spatio-temporal setting of predicting air pollution in London at hyper-local resolution.

Sections 3 and 4 introduce our shallow GP mixtures and deep GP constructions, with associated variational approximations, respectively. In Section 6 we demonstrate the empirical advantages of our framework versus the prior art followed by additional related work in Section 5 and our concluding remarks. Further analysis is provided in the Appendix and code is available at <https://github.com/ohamelijnck/multi-res-gps>.

2 Multi-resolution Multi-task Learning

Consider $\mathcal{A} \in \mathbb{N}$ observation processes $\mathbf{Y}_a \in \mathbb{R}^{N_a \times P}$ across P tasks with N_a observations. Each process may be observed at varying resolutions that arises as the volume average over a sampling area \mathcal{S}_a . Typically we discretise the area \mathcal{S}_a with a uniform grid and so we overload \mathcal{S}_a to denote these points. We construct \mathcal{A} datasets $\{(\mathbf{X}_a, \mathbf{Y}_a)\}_{a=1}^{\mathcal{A}}$, ordered by resolution size (\mathbf{Y}_1 is the highest, $\mathbf{Y}_{\mathcal{A}}$ is the lowest), where $\mathbf{X}_a \in \mathbb{R}^{N_a \times |\mathcal{S}_a| \times D_a}$ and D_a is the input dimension. For notational simplicity we assume that all tasks are observed across all processes, although this need not be the case.

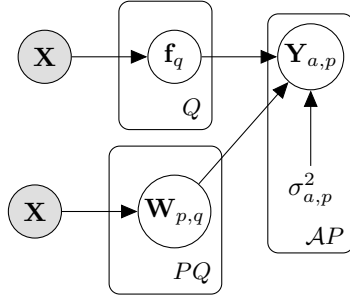
In our motivating application there are multiple sensor networks (observation processes) measuring multiple air pollutants (tasks) such as CO_2 , NO_2 , PM_{10} , $\text{PM}_{2.5}$ at different sampling resolutions. These multi-resolution observations exist both within tasks, (*intra-task multi-resolution*) when different sensor networks measure the same pollutant, and across tasks (*inter-task multi-resolution*) when different sensor networks measure different but potentially correlated pollutants due to e.g. common emission sources. Our goal is to develop scalable, non-stationary non-parametric models for air pollution while delivering accurate estimation and uncertainty quantification.

3 Multi-Resolution Gaussian Process Regression Networks (MR-GPRN)

We first introduce a *shallow* instantiation of the multi-resolution multi-task framework. MR-GPRN is a shallow GP mixture, Fig. 1, that extends the Gaussian process regression network (GPRN) [36]. Briefly, the GPRN jointly models all P tasks as a linear combination of $Q \in \mathbb{N}$ GPs. These GPs are combined using task specific weights, that are themselves GPs, resulting in $PQ \in \mathbb{N}$ latent weights $\mathbf{W}_{p,q}$. More formally, $\mathbf{f}_q \sim \mathcal{GP}(0, \mathbf{K}_q^f)$, $\mathbf{W}_{p,q} \sim \mathcal{GP}(0, \mathbf{K}_{p,q}^w)$ and each task p is modelled as $\mathbf{Y}_p = \sum_{q=1}^Q \mathbf{W}_{p,q} \odot \mathbf{f}_q + \epsilon_p$ where \odot is the Hadamard product and $\epsilon \sim \mathcal{N}(0, \sigma_p^2 \mathbf{I})$. The GPRN is an extension of the Linear Coregionalization Model (LCM) [3] and can enable the learning of non-stationary processes through input dependent weights [1].

3.1 Model Specification

We extend the GPRN model to handle multi-resolution observations by integrating the latent process over the sampling area for each observation. Apart from the standard inter-task dependency we would ideally want to be able to model additional dependencies between observation processes such as, for example, correlated noises. Directly modelling this additional dependency can quickly become intractable, due to the fact that it can vary in input space. If one ignores this dependency by assuming a product likelihood, as in [15, 19], then the misspecification results in severe posterior contractions



Algorithm 1 Inference of MR-GPRN

Input: \mathcal{A} multi-resolution datasets $\{(\mathbf{X}_a, \mathbf{Y}_a)\}_{a=1}^{\mathcal{A}}$, initial parameters θ ,
 $\hat{\theta} \leftarrow \arg \max_{\theta} \sum_{a=1}^{\mathcal{A}} \ell(\mathbf{Y}_a | \theta)$
 $\mathbf{H} \leftarrow \sum_{a=1}^{\mathcal{A}} \nabla \ell(\mathbf{Y}_a | \hat{\theta}) \nabla \ell(\mathbf{Y}_a | \hat{\theta})^T$
 $\mathbf{J} \leftarrow \nabla^2 \ell(\mathbf{Y} | \hat{\theta})$
 $\phi \leftarrow \begin{cases} \frac{|\hat{\theta}|}{\text{Tr}[\mathbf{H}(\hat{\theta})^{-1} \mathbf{J}(\hat{\theta})]} \\ \frac{\text{Tr}[\mathbf{H}(\hat{\theta}) \mathbf{J}(\hat{\theta})^{-1} \mathbf{H}(\hat{\theta})]}{\text{Tr}[\mathbf{H}(\hat{\theta})]} \end{cases}$
 $\theta \leftarrow \arg \min_{\theta} \left(\sum_{a=1}^{\mathcal{A}} \phi \mathbb{E}_q [\ell(\mathbf{Y}_a | \theta)] + \mathcal{KL} \right)$

Figure 1: **Left:** Graphical model of MR-GPRN for \mathcal{A} observation processes each with P tasks. This allows *multi-resolution learning* between and across tasks. **Right:** Inference for MR-GPRN.

(see Fig. 2). To circumvent these extremes we approximate the full likelihood using a multi-resolution composite likelihood that attempts to correct for this misspecification [33]. The posterior over the latent functions is now:

$$p(\mathbf{W}, \mathbf{f} | \mathbf{Y}) \propto \underbrace{\prod_{a=1}^{\mathcal{A}} \prod_{p=1}^P \prod_{n=1}^{N_a} \mathcal{N}(\mathbf{Y}_{a,p,n} | \frac{1}{|\mathcal{S}_{a,n}|} \int_{\mathcal{S}_{a,n}} \sum_{q=1}^Q \mathbf{W}_{p,q}(\mathbf{x}) \odot \mathbf{f}_q(\mathbf{x}) d\mathbf{x}, \sigma_{a,p}^2 \mathbf{I})}_{\text{MR-GPRN Composite Likelihood}} \underbrace{\phi}_{\text{GPRN Prior}} p(\mathbf{W}, \mathbf{f}) \quad (1)$$

where $\phi \in \mathbb{R}_{>0}$ are the composite weights that are important for inference. The integral within the multi-resolution likelihood links the underlying latent process to each of the resolutions; in general this is not available in closed form and so we approximate it by discretizing over a uniform grid. When we only have one task and \mathbf{W} is set to a vector of constants we denote the model as MR-GP.

3.2 Composite Likelihood Weights

Under a misspecified model the asymptotic distribution of the MLE estimate converges to $\mathcal{N}(\theta_0, \frac{1}{n} \mathbf{H}(\theta_0) \mathbf{J}(\theta_0)^{-1} \mathbf{H}(\theta_0))$ where θ_0 are the true parameters and $\mathbf{H}(\theta_0) = \frac{1}{n} \sum_{n=1}^N \nabla \ell(\mathbf{Y} | \theta_0) \nabla \ell(\mathbf{Y} | \theta_0)^T$, $\mathbf{J}(\theta_0) = \frac{1}{n} \sum_{n=1}^N \nabla^2 \ell(\mathbf{Y} | \theta_0)$ are the Hessian and Jacobian respectively. The form of the asymptotic variance is the *sandwich information matrix* and it represents the loss of information in the MLE estimate due to the failure of Bartlett's second identity [33].

Following Lyddon et al. [17] and Ribatet [27] we write down the asymptotic posterior of MR-GPRN as $\mathcal{N}(\theta_0, n^{-1} \phi^{-1} \mathbf{H}(\theta_0))$. In practise we only consider a subset of parameters that are present in all likelihood terms, such as the kernel parameters. Asymptotically one would expect the contribution of the prior to vanish causing the asymptotic posterior to match the limiting MLE. The composite weights ϕ can be used to bring these distributions as close together as possible. Approximating θ_0 with the MLE estimate $\hat{\theta}$ and setting $\phi^{-1} \mathbf{H}(\hat{\theta}) = \mathbf{H}(\hat{\theta}) \mathbf{J}(\hat{\theta})^{-1} \mathbf{H}(\hat{\theta})$ we can rearrange to find ϕ and recover the magnitude correction of Ribatet [27]. Instead if we take traces and then rearrange we recover the correction of Lyddon et al. [17]:

$$\phi_{\text{Ribatet}} = \frac{|\hat{\theta}|}{\text{Tr}[\mathbf{H}(\hat{\theta})^{-1} \mathbf{J}(\hat{\theta})]} \quad , \quad \phi_{\text{Lyddon}} = \frac{\text{Tr}[\mathbf{H}(\hat{\theta}) \mathbf{J}(\hat{\theta})^{-1} \mathbf{H}(\hat{\theta})]}{\text{Tr}[\mathbf{H}(\hat{\theta})]} \quad (2)$$

3.3 Inference

In this section we derive a closed form variational lower bound for MR-GPRN, the full details can be found in the Appendix. For computational efficiency we introduce inducing points (see [10, 32]) $\mathbf{U} = \{\mathbf{u}_{\mathbf{q}}\}_{q=1}^Q$ and $\mathbf{V} = \{\mathbf{v}_{\mathbf{p},\mathbf{q}}\}_{p,q=1}^{P,Q}$, for the latent GPs \mathbf{f} and \mathbf{W} respectively, where $\mathbf{u}_{\mathbf{q}} \in \mathbb{R}^M$ and $\mathbf{v}_{\mathbf{p},\mathbf{q}} \in \mathbb{R}^M$. The inducing points are at the corresponding locations $\mathbf{Z}^{(\mathbf{u})} = \{\mathbf{Z}_{\mathbf{q}}^{(\mathbf{u})}\}_{q=1}^Q$, $\mathbf{Z}^{(\mathbf{v})} =$

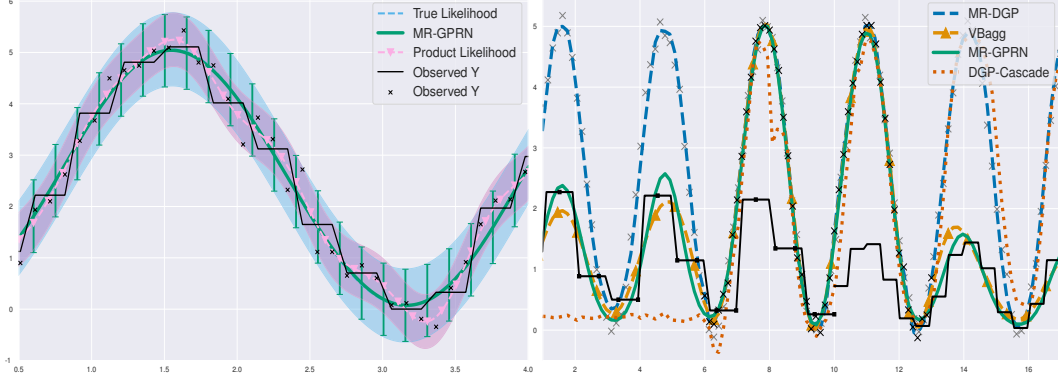


Figure 2: **Left:** MR-GPRN recovers the true predictive variance whereas assuming a product likelihood assumption leads to posterior contraction. **Right:** MR-DGP recovers the true predictive mean under a multi-resolution setting with scaling biases. Both VBAGG-NORMAL and MR-GPRN fail as they propagate the bias. Black crosses and lines denote observed values. Grey crosses denote observations removed for testing.

$\{\mathbf{Z}_{\mathbf{p},\mathbf{q}}^{(\mathbf{v})}\}_{p,q=1}^{P,Q}$ for $\mathbf{Z}^{(\cdot)} \in \mathbb{R}^{M,D}$. We construct the augmented posterior and use the approximate posterior $q(\mathbf{u}, \mathbf{v}, \mathbf{f}, \mathbf{W}) = p(\mathbf{f}, \mathbf{W} | \mathbf{u}, \mathbf{v})q(\mathbf{u}, \mathbf{v})$ where

$$q(\mathbf{u}, \mathbf{v}) = \sum_{k=1}^K \pi_k \prod_{j=1}^Q \mathcal{N}(\mathbf{m}_{k,j}^{(\mathbf{u})}, \mathbf{S}_{k,j}^{(\mathbf{u})}) \cdot \prod_{i,j=1}^{P,Q} \mathcal{N}(\mathbf{m}_{k,i,j}^{(\mathbf{v})}, \mathbf{S}_{k,i,j}^{(\mathbf{v})}) \quad (3)$$

is a free form mixture of Gaussians with K components. We follow the variational derivation of [14, 22] and derive our expected log-likelihood $\text{ELL} = \sum_{a=1}^A \sum_{p=1}^P \sum_{n=1}^{N_a} \sum_{k=1}^K \text{ELL}_{a,p,n,k}$,

$$\begin{aligned} \text{ELL}_{a,p,n,k} = & \pi_k \log \mathcal{N} \left(Y_{a,p,n} \mid \frac{1}{|\mathcal{S}_{a,n}|} \sum_{\mathbf{x} \in \mathcal{S}_{a,n}} \sum_{q=1}^Q \boldsymbol{\mu}_{k,p,q}^{(w)}(\mathbf{x}) \boldsymbol{\mu}_{k,q}^{(f)}(\mathbf{x}), \sigma_{a,p}^2 \right) \\ & - \frac{\pi_k}{2\sigma_{a,p}^2} \frac{1}{|\mathcal{S}_{a,n}|^2} \sum_{q=1}^Q \sum_{\mathbf{x}_1, \mathbf{x}_2} \boldsymbol{\Sigma}_{k,p,q}^{(w)} \boldsymbol{\Sigma}_{k,q}^{(f)} + \boldsymbol{\mu}_{k,q}^{(f)}(\mathbf{x}_1) \boldsymbol{\Sigma}_{k,p,q}^{(w)} \boldsymbol{\mu}_{k,q}^{(f)}(\mathbf{x}_2) \boldsymbol{\mu}_{k,p,q}^{(w)}(\mathbf{x}_1) \boldsymbol{\Sigma}_{k,q}^{(f)} \boldsymbol{\mu}_{k,p,q}^{(w)}(\mathbf{x}_2) \end{aligned} \quad (4)$$

where $\boldsymbol{\Sigma}_{k,q}^{(\cdot)}$ is evaluated at the points $\mathbf{x}_1, \mathbf{x}_2$. and $\boldsymbol{\mu}_{k,q}^{(f)}, \boldsymbol{\mu}_{k,p,q}^{(w)}, \boldsymbol{\Sigma}_{k,q}^{(f)}, \boldsymbol{\Sigma}_{k,p,q}^{(w)}$ are respectively the mean and variance of $q_k(\mathbf{W}_p), q_k(\mathbf{f})$. To infer the composite weights we follow [17, 27] and first obtain the MLE estimate of θ by maximizing the likelihood in Eq. 1. The weights can then be calculated and the variational lowerbound optimised as in Alg. 1 with $\mathcal{O}(E \cdot (PQ + Q)NM^2)$ for E optimization steps until convergence. Our closed form ELBO generalizes prior state of the art of the GPRN ([1, 14, 20]) by extending to the multi-resolution setting and allowing for a free form mixture of Gaussians variational posterior. In the Appendix we also provide variational lower bounds for the positively-restricted GPRN form $\mathbf{Y}_p = \sum_{q=1}^Q \exp(\mathbf{W}_{p,q}) \odot \mathbf{f}_q + \epsilon$ that improves identifiability and predictive performance.

3.4 Prediction

Although the full predictive distribution of a specific observation process is not available in closed form, using the variational posterior we derive the predictive mean and variance, avoiding Monte Carlo estimates. The mean is simply $\mathbb{E}[\mathbf{Y}_{a,p}^*] = \sum_k^K \pi_k \mathbb{E}_k[\mathbf{W}_{k,p}^*] \mathbb{E}_k[\mathbf{f}_k^*]$, where K is the number of components in the mixture of Gaussians variational posterior and π_k is the k 'th weight. We provide the predictive variance and full derivations in the appendix .

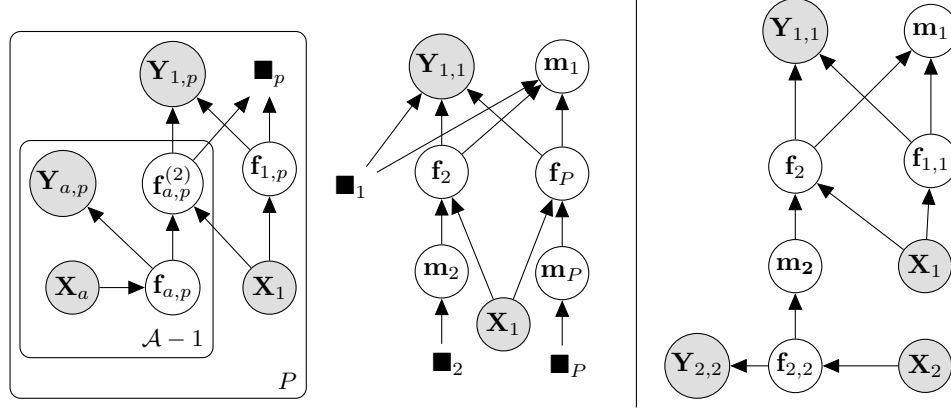


Figure 3: **Left:** General plate diagram of MR-DGP for \mathcal{A} observation processes across P tasks with noise variances omitted. For notational simplicity we have assumed that the target resolution is $a = 1$ and we use \blacksquare_p to depict each of the sub-plate diagrams defined on the LHS. **Right:** A specific instantiation of an MR-DGP for 2 tasks and 2 observation processes (resolutions) with a target process $Y_{1,1}$ as in the *inter*-task multi-resolution PM10, PM25 experiment in Section 4.

4 Multi-Resolution Deep Gaussian Processes (MR-DGP)

We now introduce MR-DGP, a deep instantiation of the framework which extends the deep GP (DGP) model of Damianou and Lawrence [5] into a tree-structured multi-resolution construction, Fig. 3. For notational convenience henceforth we assume that $p = 1$ is the target task and that $a = 1$ is the highest resolution and the one of primary interest. We note that this need not be the case and the relevant expressions can be trivially updated accordingly.

4.1 Model Specification

First we focus on the case when $P = 1$ and then generalize to an arbitrary number of tasks. We place \mathcal{A} independent “Base” GPs $\{f_{a,p}\}_{a=1}^{\mathcal{A}}$ on each of the \mathcal{A} datasets within task p that model their corresponding resolution independently. Taking $a = 1$ to be the target observation process we now construct $\mathcal{A} - 1$ two-layer DGPs that map from these base GPs $\{f_{a,p}\}_{a=2}^{\mathcal{A}}$ to the target process $a = 1$ while learning an input-dependent mapping between observation processes. These DGPs are local experts that capture the information contained in each resolution for the target observation process. Every GP has an explicit likelihood which enables us to estimate and predict at every resolution and task while allowing for biases between observation processes to be corrected, see Fig. 2.

More formally, the joint distribution $p(\mathbf{Y}_p, \mathbf{F}_p)$ is given by:

$$\underbrace{\prod_{a=2}^{\mathcal{A}} \mathcal{N}(\mathbf{Y}_{1,p} | \frac{1}{|S_a|} \int_{S_a} \mathbf{f}_{a,p}^{(2)}(\mathbf{x}) d\mathbf{x}, \sigma_{a,p}^2) p(\mathbf{f}_{a,p}^{(2)} | \mathbf{f}_{a,p})}_{\text{Deep GPs}} \underbrace{\prod_{a=1}^{\mathcal{A}} \mathcal{N}(\mathbf{Y}_{a,p} | \frac{1}{|S_a|} \int_{S_a} \mathbf{f}_{a,p}(\mathbf{x}) d\mathbf{x}, \sigma_{a,p}^2) p(\mathbf{f}_{a,p})}_{\text{Base GPs}} \quad (5)$$

where $\mathbf{f}_{a,p} \sim \mathcal{GP}(0, \mathbf{K}_{a,p})$ and we have stacked all the observations and latent GPs into \mathbf{Y}_p and \mathbf{F}_p respectively. Each of the likelihood components is a special case of the multi-resolution likelihood in Eq. 1 (where $Q = 1$ and the latent GPs \mathbf{W} are constant) and we discretize the integral in the same fashion. Similarly to the deep multi-fidelity model of [4] we define each DGP as:

$$p(\mathbf{f}_{a,p}^{(2)} | \mathbf{f}_{a,p}) = \mathcal{N}(0, \mathbf{K}_{a,p}^{(2)}((\mathbf{f}_{a,p}, \mathbf{X}_1), (\mathbf{f}_{a,p}, \mathbf{X}_1))) \quad (6)$$

where \mathbf{X}_1 are the covariates of the resolution of interest in our running example and allow each DGP to learn a mapping, between any observation process a and the target one, that varies across \mathbf{X}_1 . We now have \mathcal{A} independent DGPs modelling $\mathbf{Y}_{1,p}$ with separable spatio-temporal kernels at each layer. The observation processes are not only at varying resolutions, but could also be partially overlapping or disjoint. This motivates treating each GP as a local model in a mixture of GP experts [38]. Mixture of GP experts typically combine the local GPs in two ways: either through a gating

network [25] or through weighing the local GPs [6, 21]. We employ the mixing weight approach in order to avoid the computational burden of learning the gating work. We define the mixture $\mathbf{m}_p = \beta_1 \odot \mathbf{f}_{1,p} + \sum_{a=1}^A \beta_a \odot \mathbf{f}_{a,p}^{(2)}$ where the weight captures the reliability of the local GPs (or is set to 1 if the mixture is a singleton). The reliability is defined by the resolution and support of the base GPs and is naturally achieved by utilising the normalised log variances of the base GPs as $\beta_a = (1 - \mathbf{V}_a) \sum_i^a \mathbf{V}_i$. We provide the full justification and derivation for these weights in the appendix.

We can now generalize to an arbitrary number of tasks. For each task we construct a mixture of experts \mathbf{m}_p as described above. For tasks $p > 1$ we learn the mapping from \mathbf{m}_p to the target observation process $\mathbf{Y}_{1,1}$. This defines another set of local GP experts that is combined into a mixture with DGP experts. In our experiments we set \mathbf{m}_p for $p > 1$ to be a simple average and for \mathbf{m}_1 we use our variance derived weights. This formulation naturally handles biases between the mean of different observations processes and each layer of the DGPs has a meaningful interpretation as it is modelling a specific observation process.

4.2 Augmented Posterior

Due to the non-linear forms of the parent GPs within the DGPs, marginalising out the parent GPs is generally analytically intractable. Following [28] we introduce inducing points $\mathbf{U} = \{\mathbf{u}_p\}_{p=2}^P \cup \{\mathbf{u}_{a,p}^{(2)}, \mathbf{u}_{a,p}\}_{a,p=1}^{P,A}$ where each $\mathbf{u}_{\cdot,\cdot}^{(\cdot)} \in \mathbb{R}^M$ and inducing locations $\mathbf{Z} = \{\mathbf{Z}_p\}_{p=2}^P \cup \{\mathbf{Z}_{a,p}^{(2)}, \mathbf{Z}_{a,p}\}_{a,p=1}^{P,A}$ where $\mathbf{Z}_p, \mathbf{Z}_{a,p}^{(2)} \in \mathbb{R}^{M \times (D+1)}$ and $\mathbf{Z}_{a,p} \in \mathbb{R}^{M \times D}$. The augmented posterior is now simply $p(\mathbf{Y}, \mathbf{F}, \mathbf{M}, \mathbf{U}) = p(\mathbf{Y}|\mathbf{F})p(\mathbf{F}, \mathbf{M}|\mathbf{U})p(\mathbf{U})$ where each $p(\mathbf{u}_{\cdot,\cdot}^{(\cdot)}) = \mathcal{N}(0, \mathbf{K}_{\cdot,\cdot}^{(\cdot)})$. Full details are provided in the appendix.

4.3 Inference

Following [28] we construct an approximate augmented posterior that maintains the dependency structure between layers:

$$q(\mathbf{M}, \mathbf{F}, \mathbf{U}) = p(\mathbf{M}, \mathbf{F}|\mathbf{U}) \prod_{p=2}^P q(\mathbf{u}_p) \cdot \prod_{p=1}^P \prod_{a=1}^A q(\mathbf{u}_{a,p}^{(2)})q(\mathbf{u}_{a,p}) \quad (7)$$

where each $q(\mathbf{u}_{\cdot,\cdot}^{(\cdot)})$ are independent free-form Gaussians $\mathcal{N}(\mathbf{m}_{\cdot,\cdot}^{(\cdot)}, \mathbf{S}_{\cdot,\cdot}^{(\cdot)})$ and the conditional is

$$p(\mathbf{F}, \mathbf{M}|\mathbf{U}) = \prod_{p=2}^P p(\mathbf{f}_p|\mathbf{m}_p, \mathbf{u}_p)p(\mathbf{m}_p|\text{Pa}(\mathbf{m}_p)) \cdot \prod_{p=1}^P p(\mathbf{f}_{1,p}|\mathbf{u}_{1,p}) \prod_{a=2}^A p(\mathbf{f}_{a,p}^{(2)}|\mathbf{f}_{a,p}, \mathbf{u}_{a,p}^{(2)})p(\mathbf{f}_{a,p}|\mathbf{u}_{a,p}). \quad (8)$$

We use $\text{Pa}(\cdot)$ to denote the set of parent GPs of a given GP and $\mathcal{L}(\mathbf{f})$ to denote the depth of DGP \mathbf{f} , $p(\mathbf{m}_p|\text{Pa}(\mathbf{m}_p)) = \mathcal{N}(\sum_a^A \mathbf{w}_{a,p} \mu_{a,p}, \sum_a^A \mathbf{w}_{a,p} \Sigma_{a,p} \mathbf{w}_{a,p})$ and $\mu_{a,p}, \Sigma_{a,p}$ are the means and variances of the relevant DGPs. Note that the mixture \mathbf{m}_1 combines all the DGPs at the top layer of the tree-hierarchy and hence it only appears in the predictive distribution of MR-DGP. All other terms are standard sparse GP conditionals and are provided in the Appendix. The ELBO is then simply derived as

$$\mathcal{L}_{\text{MR-DGP}} = \underbrace{\mathbb{E}_{q(\mathbf{M}, \mathbf{F}, \mathbf{U})} [\log p(\mathbf{Y}|\mathbf{F})]}_{\text{ELL}} + \underbrace{\mathbb{E}_{q(\mathbf{U})} \left[\log \frac{P(\mathbf{U})}{q(\mathbf{U})} \right]}_{\text{KL}} \quad (9)$$

where the KL term is decomposed into a sum over all inducing variables $\mathbf{u}_{\cdot,\cdot}^{(\cdot)}$. The expected log likelihood (ELL) term decomposes across all \mathbf{Y} :

$$\sum_{p=2}^P \mathbb{E}_{q(\mathbf{f}_p)} [\log p(\mathbf{Y}_{1,1}|\mathbf{f}_p)] + \sum_{p=1}^P \sum_a^A \left[\mathbb{E}_{q(\mathbf{f}_{a,1}^{(2)})} [\log p(\mathbf{Y}_{1,1}|\mathbf{f}_{a,1}^{(2)})] + \mathbb{E}_{q(\mathbf{f}_{a,p})} [\log p(\mathbf{Y}_{a,p}|\mathbf{f}_{a,p})] \right]. \quad (10)$$

For each ELL component the marginal $q(\mathbf{f}_{\cdot,\cdot}^{(\cdot)})$ is required. Because the base GPs are Gaussian, sampling is straightforward and the samples can be propagated through the layers, allowing the marginalization integral to be approximated by Monte Carlo samples. We use the reparametrization trick to draw samples from the variational posteriors [11]. The inference procedure is given in Alg. 2.

Algorithm 2 Inference procedure for MR-DGP

Input: P multi-resolution datasets $\{(\mathbf{X}_p, \mathbf{Y}_p)\}_{p=1}^P$, initial parameters θ_0 ,
procedure MARGINAL($\mathbf{f}, \mathbf{X}, l, L$)
 if $l = L$ **then**
 return $q(\mathbf{f}|\mathbf{X})$
 end if
 $q(\mathcal{P}(\mathbf{f})|\mathbf{X}) \leftarrow \text{MARGINAL}(\mathcal{P}(\mathbf{f}), \mathbf{X}, l+1, L(\mathcal{P}(\mathbf{f})))$
 return $\frac{1}{S} \sum_{s=1}^S p(\mathbf{f}|\mathbf{f}^{(s)}, \mathbf{X})$ where $\mathbf{f}^{(s)} \sim q(\mathcal{P}(\mathbf{f})|\mathbf{X})$
end procedure
 $\theta \leftarrow \arg \min_{\theta} \left[\mathbb{E}_{\{\text{MARGINAL}(\mathbf{f}_p, \mathbf{X}_p, 0, L(\mathbf{f}_p))\}_{p=1}^P} [\log p(\mathbf{Y}|\mathbf{F}, \mathbf{X}, \theta)] + \mathcal{KL}(q(\mathbf{U})||p(\mathbf{U})) \right]$

4.4 Prediction

Predictive Density. To predict at $\mathbf{x}^* \in \mathbb{R}^D$ in the target resolution $a = 1$ we simply approximate the predictive density $q(\mathbf{m}_1^*)$ by sampling from the variational posteriors and propagating the samples $\mathbf{f}^{(s)}$ through all the layers of the MR-DGP structure:

$$q(\mathbf{m}_1^*) = \int q(\mathbf{m}_1^*|\text{Pa}(\mathbf{m}_1^*)) \prod_{\mathbf{f} \in \text{Pa}(\mathbf{m}_1^*)} q(\mathbf{f}) d\text{Pa}(\mathbf{m}_1^*) \approx \frac{1}{S} \sum_{s=1}^S q(\mathbf{m}_1^*|\{\mathbf{f}^{(s)}\}_{\mathbf{f} \in \text{Pa}(\mathbf{m}_1^*)}) \quad (11)$$

In fact while propagating the samples through the tree structure the model naturally predicts at every resolution a and task p for the corresponding input location.

5 Related Work

Gaussian processes (GPs) are the workhorse for spatio-temporal modelling in spatial statistics [9] and in machine learning [26] with the direct link between multi-task GPs and Linear Models of Coregionalisation (LCM) reviewed by Alvarez et al. [3]. Heteroscedastic GPs [16] and recently proposed deeper compositions of GPs for the multi-fidelity setting [4, 23, 24] assume that all observations are of the same resolution. In spatial statistics the related *change of support* problem has been approached through Markov Chain Monte Carlo approximations and domain discretizations [8, 9]. Concurrently to our work [39] has explored the change of support problem under the setting of multi-variate areal data. They do not consider sparse GPs and hence can derive the true GP posterior, that is then approximated through an integral discretisation. A recent exception to discretising the multi-resolution integral is the work by Smith et al. [30] that solves the integral for squared exponential kernels but only considers observations from one resolution and cannot handle additional input features. Independently, and concurrently, [37] have recently proposed a multi-resolution LCM model that is similar to our MR-GPRN model without dependent observation processes and composite likelihood corrections. Instead they focus on improved estimation of the area integral and non-Gaussian likelihoods. Finally, we note that the multi-resolution GP work by Fox and Dunson [7] defines a DGP construction for non-stationary models that is more akin to multi-scale modelling [35]. This line of research typically focuses on learning multiple kernel lengthscales to explain both broad and fine variations in the underlying process and hence cannot handle multi-resolution observations.

6 Experiments

We demonstrate and evaluate the MRGPs on synthetic experiments and the challenging problem of estimating and forecasting air pollution in the city of London. We compare against VBAGG-NORMAL [15] and two additional baselines. The first, CENTER-POINT, is a GPRN modified to support multi-resolution data by representing each aggregation region through its centre point only. The second, MR-CASCADE is an instance of MR-DGP but, to illustrate the benefits of the tree composition and the mixture of experts approach of MR-DGP, instead of a tree structured DGP (as in Fig. 3) we

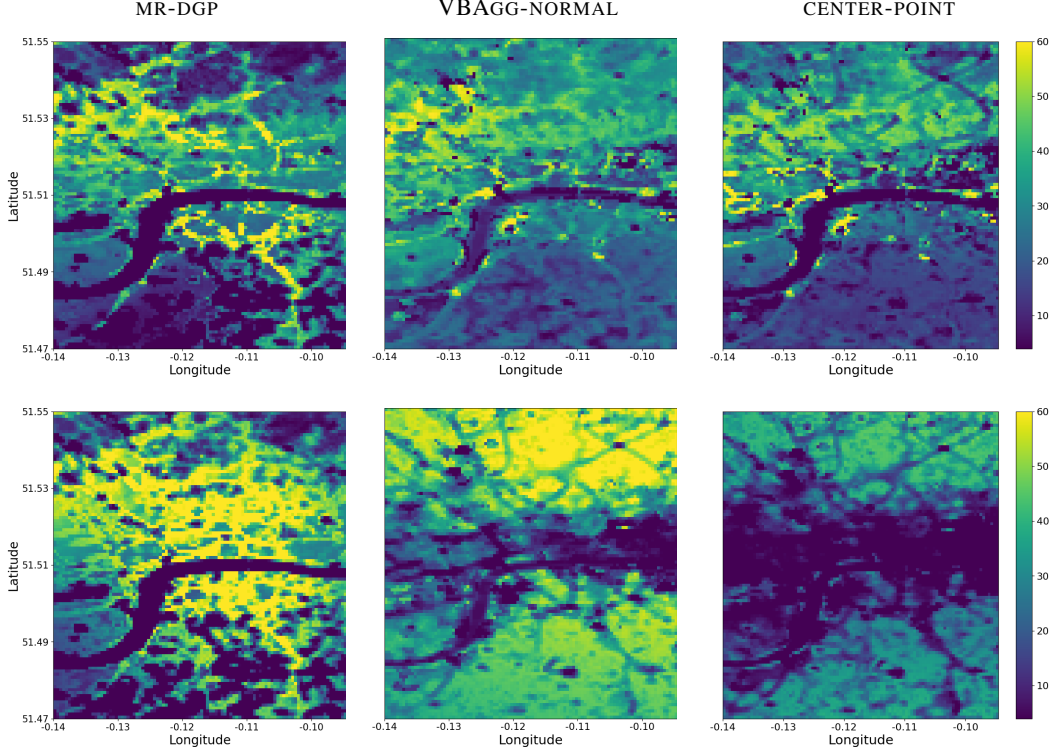


Figure 4: Spatio-temporal estimation and forecasting of NO_2 levels in London. **Top Row:** Spatial slices from MR-GPRN, VBAGG-NORMAL and CENTER-POINT respectively at 19/02/2019 11:00:00 using observations from both LAQN and the satellite model (low spatial resolution). **Bottom Row:** Spatial slices at the base resolution from the same models at 19/02/2019 17:00:00 where *only* observations from the satellite model are present.

construct a cascade. Experiments are coded¹ in *TensorFlow* and we provide additional analysis in the Appendix.

Dependent observation processes: We provide details of the dependent observation processes experiment in the left of Fig. 2 in the Appendix.

Biased observation processes: To demonstrate the ability of MR-DGP in handling biases across observation processes we construct 3 datasets from the function $y = s \cdot 5 \sin(x)^2 + 0.1\epsilon$ where $\epsilon \sim \mathcal{N}(0, 1)$. The first $\mathbf{X}_1, \mathbf{Y}_1$ is at resolution $S_1 = 1$ in the range $x=[7, 12]$ with a scale $s = 1$. The second is at resolution of $S_2 = 5$ between $x=[-10, 10]$ with a scale $s = 0.5$ and lastly the third is at resolution of $S_3 = 5$ $x=[10, 20]$ with a scale $s = 0.3$. The aim is to predict y across the range $[-10, 20]$ and the results are shown in Table 2 and Fig. 2. MR-DGP significantly outperforms all of the four alternative approaches as it is learning a forward *mapping* between observations.

Training. When training both MR-GPRN and VBAGG-NORMAL we first jointly optimize the variational and hyper parameters while keeping the likelihood variances fixed and then jointly optimize all parameters together. For MR-DGP we first optimize layer by layer and then jointly optimize all parameters together, see appendix, as we find that this helps to avoid early local optima.

Inter-task multi-resolution: modelling of PM_{10} and PM_{25} in London: In this experiment we consider multiple tasks with different resolutions. We jointly model PM_{10} and PM_{25} at a specific LAQN location in London. The site we consider is *RB7* in the date range 18/06/2018 to 28/06/2018. At this location we have hourly data from both PM_{10} and PM_{25} . To simulate having multiple resolutions we construct 2, 5, 10 and 24 hour aggregations of PM_{10} and remove a 2 day region of

¹Codebase and datasets to reproduce results are available at https://github.com/ohameliijnck/multi_res_gps

Table 1: *Inter*-task multi-resolution. Missing data predictive MSE on PM_{25} from MR-GPRN, MR-DGP and baseline CENTER-POINT for 4 different aggregation levels of PM_{10} . VBAGG-NORMAL is inapplicable in this experiment as it is a single-task approach.

Model	PM ₁₀ Resolution			
	2 Hours	5 Hours	10 Hours	24 Hours
CENTER-POINT	4.67 \pm 0.74	5.04 \pm 0.45	5.26 \pm 0.91	5.72 \pm 0.91
MR-GPRN	4.54 \pm 0.93	5.09 \pm 1.04	4.96 \pm 1.07	5.32 \pm 1.14
MR-DGP	5.14 \pm 1.28	4.81 \pm 1.06	4.61 \pm 1.43	5.42 \pm 1.15

Table 2: *Intra*-task multi-resolution. **Left:** Predicting NO₂ across London (Fig. 4). **Right:** Synthetic experiment results (Fig. 2) with three observations processes and scaling bias.

Model	RMSE	MAPE	Model	RMSE	MAPE
Single GP	20.55 \pm 9.44	0.8 \pm 0.16	MR-CASCADE	2.12	0.16
CENTER-POINT	18.74 \pm 12.65	0.65 \pm 0.21	VBAGG-NORMAL	1.68	0.14
VBAGG-NORMAL	16.16 \pm 9.44	0.69 \pm 0.37	MR-GPRN	1.6	0.14
MR-GPRN w/o CL	12.97 \pm 9.22	0.56 \pm 0.32	MR-DGP	0.19	0.02
MR-GPRN w CL	11.92 \pm 6.8	0.45 \pm 0.17			
MR-DGP	6.27 \pm 2.77	0.38 \pm 0.32			

PM_{25} which is the test region. The results from all of our models in Table 1 demonstrate the ability to successfully learn the multi-task dependencies. Note that CENTER-POINT fails, e.g. Table 2, when the sampling area cannot be approximated by a single center point due to the scale of the underlying process.

***Intra*-task multi-resolution: spatio-temporal modelling of NO₂ in London:** In this experiment we consider the case of a single task but with multiple multi-resolution observation processes. First we use observations coming from ground point sensors from the London Air Quality Network (LAQN). These sensors provide hourly readings of NO₂. Secondly we use observations arising from a global satellite model [18] that provide hourly data at a spatial resolution of 7km \times 7km and provide 48 hour forecasts. We train on both the LAQN and satellite observations from 19/02/2018-20/02/2018 and the satellite ones from 20/02/2018-21/02/2018. We then predict at the resolution of the LAQN sensors in the latter date range. To calculate errors we predict for each LAQN sensor site, and find the average and standard deviation across all sites.

We find that MR-DGP is able to substantially outperform both VBAGG-NORMAL, MR-GPRN and the baselines, Table 2 (left), as it is learning the forward mapping between the low resolution satellite observations and the high resolution LAQN sensors, while handling scaling biases. This is further highlighted in the bottom of Fig. 4 where MR-DGP is able to retain high resolution structure based only on satellite observations whereas VBAGG-NORMAL and CENTER-POINT over-smooth.

7 Conclusion

We offer a framework for evidence integration when observation processes can have varying *inter*- and *intra*-task sampling resolutions, dependencies, and different signal to noise ratios. Our motivation comes from a challenging and impactful problem of hyper-local air quality prediction in the city of London, while the underlying multi-resolution multi-sensor problem is general and pervasive across modern spatio-temporal settings and applications of machine learning. We proposed both shallow mixtures and deep learning models that generalise and outperform the prior art, correct for posterior contraction, and can handle biases in observation processes such as discrepancies in the mean. Further directions now open up to robustify the multi-resolution framework against outliers and against further model misspecification by exploiting ongoing advances in generalized variational inference [12]. Finally an open challenge remains on developing continuous model constructions that avoid domain discretization, as in [2, 37], for more complex settings.

Acknowledgements

O. H., T. D and K.W. are funded by the Lloyd’s Register Foundation programme on Data Centric Engineering through the London Air Quality project. This work is supported by The Alan Turing Institute for Data Science and AI under EPSRC grant EP/N510129/1 in collaboration with the Greater London Authority. We would like to thank the anonymous reviewers for their feedback and Libby Rogers, Patrick O’Hara, Daniel Tait and Juan Maroñas for their help on multiple aspects of this work.

References

- [1] (2008). Gaussian process product models for nonparametric nonstationarity. In *Proceedings of the 25th International Conference on Machine Learning*.
- [2] Adelsberg, M. and Schwantes, C. (2018). Binned kernels for anomaly detection in multi-timescale data using Gaussian processes. In *Proceedings of the KDD 2017: Workshop on Anomaly Detection in Finance*, Proceedings of Machine Learning Research.
- [3] Alvarez, M. A., Rosasco, L., Lawrence, N. D., et al. (2012). Kernels for vector-valued functions: A review. *Foundations and Trends® in Machine Learning*, 4(3):195–266.
- [4] Cutajar, K., Pullin, M., Damianou, A., Lawrence, N., and González, J. (2019). Deep Gaussian Processes for Multi-fidelity Modeling. *arXiv e-prints*, page arXiv:1903.07320.
- [5] Damianou, A. and Lawrence, N. (2013). Deep Gaussian processes. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*.
- [6] Deisenroth, M. P. and Ng, J. W. (2015). Distributed gaussian processes. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML’15, pages 1481–1490. JMLR.org.
- [7] Fox, E. B. and Dunson, D. B. (2012). Multiresolution Gaussian processes. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*.
- [8] Fuentes, M. and Raftery, A. E. (2005). Model evaluation and spatial interpolation by Bayesian combination of observations with outputs from numerical models. *Biometrics*.
- [9] Gelfand, A., Fuentes, M., Guttorp, P., and Diggle, P. (2010). *Handbook of Spatial Statistics*. Chapman & Hall/CRC Handbooks of Modern Statistical Methods. Taylor & Francis.
- [10] Hensman, J., Fusi, N., and Lawrence, N. D. (2013). Gaussian processes for big data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*.
- [11] Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In *International Conference for Learning Representations*.
- [12] Knoblauch, J., Jewson, J., and Damoulas, T. (2019). Generalized Variational Inference. *arXiv e-prints*, page arXiv:1904.02063.
- [13] Krause, A., Singh, A., and Guestrin, C. (2008). Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies.
- [14] Krauth, K., Bonilla, E. V., Cutajar, K., and Filippone, M. (2017). AutoGP: Exploring the Capabilities and Limitations of Gaussian Process Models. In *Conference on Uncertainty in Artificial Intelligence (UAI)*.
- [15] Law, H. C. L., Sejdinovic, D., Cameron, E., Lucas, T. C., Flaxman, S., Battle, K., and Fukumizu, K. (2018). Variational learning on aggregate outputs with Gaussian processes. *Advances in Neural Information Processing Systems (NeurIPS)*.
- [16] Lázaro-Gredilla, M. and Titsias, M. K. (2011). Variational heteroscedastic Gaussian process regression. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*.

- [17] Lyddon, S. P., Holmes, C. C., and Walker, S. G. (2019). General Bayesian updating and the loss-likelihood Bootstrap. *Biometrika*.
- [18] Marécal, V., Peuch, V.-H., Andersson, C., Andersson, S., Arteta, J., Beekmann, M., Benedictow, A., Bergström, R., Bessagnet, B., Cansado, A., Chéroux, F., Colette, A., Coman, A., Curier, R. L., Denier van der Gon, H. A. C., Drouin, A., Elbern, H., Emili, E., Engelen, R. J., Eskes, H. J., Foret, G., Friese, E., Gauss, M., Giannaros, C., Guth, J., Joly, M., Jaumouillé, E., Josse, B., Kadyrov, N., Kaiser, J. W., Krajsek, K., Kuenen, J., Kumar, U., Liora, N., Lopez, E., Malherbe, L., Martinez, I., Melas, D., Meleux, F., Menut, L., Moinat, P., Morales, T., Parmentier, J., Piacentini, A., Plu, M., Poupkou, A., Queguiner, S., Robertson, L., Rouil, L., Schaap, M., Segers, A., Sofiev, M., Tarasson, L., Thomas, M., Timmermans, R., Valdebenito, A., van Velthoven, P., van Versendaal, R., Vira, J., and Ung, A. (2015). A regional air quality forecasting system over europe: the macc-ii daily ensemble production. *Geoscientific Model Development*.
- [19] Moreno-Muñoz, P., Artés-Rodríguez, A., and Álvarez, M. A. (2018). Heterogeneous multi-output Gaussian process prediction. In *Proceedings of the 32Nd International Conference on Neural Information Processing Systems*.
- [20] Nguyen, T. and Bonilla, E. (2013). Efficient variational inference for Gaussian process regression networks. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*.
- [21] Nguyen, T. and Bonilla, E. (2014a). Fast allocation of Gaussian process experts. In *Proceedings of the 31st International Conference on Machine Learning*.
- [22] Nguyen, T. V. and Bonilla, E. V. (2014b). Automated variational inference for Gaussian process models. In *Advances in Neural Information Processing Systems 27*.
- [23] Perdikaris, P., Raissi, M., Damianou, A., D. Lawrence, N., and Karniadakis, G. (2017). Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science*.
- [24] Perdikaris, P., Venturi, D., Royset, J. O., and Karniadakis, G. E. (2015). Multi-fidelity modelling via recursive co-kriging and Gaussian-markov random fields. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*.
- [25] Rasmussen, C. E. and Ghahramani, Z. (2002). Infinite mixtures of Gaussian process experts. In *Advances in Neural Information Processing Systems 14*.
- [26] Rasmussen, C. E. and Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press.
- [27] Ribatet, M. (2012). Bayesian inference from composite likelihoods, with an application to spatial extremes. In *Statistica Sinica 22: 813–845*.
- [28] Salimbeni, H. and Deisenroth, M. (2017). Doubly stochastic variational inference for deep Gaussian processes. In *Advances in Neural Information Processing Systems 30*.
- [29] Serban, I. V., Klinger, T., Tesauro, G., Talamadupula, K., Zhou, B., Bengio, Y., and Courville, A. (2017). Multiresolution recurrent neural networks: An application to dialogue response generation. In *Thirty-First AAAI Conference on Artificial Intelligence*.
- [30] Smith, M. T., Alvarez, M. A., and Lawrence, N. D. (2018). Gaussian process regression for binned data. *arXiv e-prints*.
- [31] Stoeck, J. and Friel, N. (2015). Calibration of conditional composite likelihood for Bayesian inference on Gibbs random fields. In *Artificial Intelligence and Statistics*.
- [32] Titsias, M. (2009). Variational learning of inducing variables in sparse Gaussian processes. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*.
- [33] Varin, C., Reid, N., and Firth, D. (2011a). An overview of composite likelihood methods. *Statist. Sinica*.

- [34] Varin, C., Reid, N., and Firth, D. (2011b). An overview of composite likelihood methods. *Statistica Sinica*.
- [35] Walder, C., Kim, K. I., and Schölkopf, B. (2008). Sparse multiscale Gaussian process regression. In *Proceedings of the 25th international conference on Machine learning*.
- [36] Wilson, A. G., Knowles, D. A., and Ghahramani, Z. (2012). Gaussian process regression networks. In *Proceedings of the 29th International Conference on Machine Learning*.
- [37] Yousefi, F., Smith, M. T., and Alvarez, M. A. (2019). Multi-task learning for aggregated data using gaussian processes. In *NeurIPS, 2019 (to appear)*.
- [38] Yuan, C. and Neubauer, C. (2009). Variational mixture of Gaussian process experts. In *Advances in Neural Information Processing Systems 21*.
- [39] Yusuke Tanaka, Toshiyuki Tanaka, T. I. T. K. M. O. Y. A. H. T. (2019). Spatially aggregated gaussian processes with multivariate areal outputs. In *NeurIPS, 2019 (to appear)*.

A MR-GPRN

In this section we provide the full derivation of the variational lower bound for MR-GPRN. Recall that we have a multi-resolution data set across P tasks and we model each task as

$$\mathbf{Y}_{a,p} = \frac{1}{|\mathcal{S}_a|} \int_{\mathcal{S}_a} \sum_{q=1}^Q \mathbf{W}_{p,q}(\mathbf{x}) \mathbf{f}_q(\mathbf{x}) d\mathbf{x} + \epsilon_{a,p} \quad (12)$$

where $\epsilon_{a,p} \sim \mathcal{N}(0, \sigma_{a,p}^2 \mathbf{I})$. The joint density $p(\mathbf{Y}, \mathbf{W}, \mathbf{f})$ is then proportional to

$$\underbrace{\prod_{a=1}^A \prod_{p=1}^P \prod_{n=1}^{N_a} \mathcal{N}(\mathbf{Y}_{a,p,n} | \frac{1}{|\mathcal{S}_{a,n}|} \int_{\mathcal{S}_{a,n}} \sum_{q=1}^Q \mathbf{W}_{p,q}(\mathbf{x}) \odot \mathbf{f}_q(\mathbf{x}) d\mathbf{x}, \sigma_{a,n}^2 \mathbf{I})^\phi}_{\text{MR-GPRN Composite Likelihood}} \underbrace{\prod_{p=1}^P \prod_{q=1}^Q p(\mathbf{W}_{a,p}) \prod_{q=1}^Q p(\mathbf{f}_q)}_{\text{GPRN Prior}} \quad (13)$$

where $\mathbf{f}_q \sim \mathcal{N}(0, \mathbf{K}_q)$ are global functions across all tasks and $\mathbf{W}_{p,q} \sim \mathcal{N}(0, \mathbf{K}_{p,q})$ are task specific. To allow for computationally efficient inference we introduce inducing points for all latent functions [32]. For the latent basis functions, \mathbf{f} , we have the inducing points $\mathbf{u} = \{\mathbf{u}_q\}_{q=1}^Q$ where $\mathbf{u}_q \in \mathbb{R}^{M_q}$ at locations $\mathbf{Z}^{(f)} = \{\mathbf{Z}_q\}_{q=1}^Q$ for $\mathbf{Z}_q \in \mathbb{R}^{M_q, D}$. Similarly, for the latent weight functions, \mathbf{W} , we have $\mathbf{v} = \{\mathbf{v}_{p,q}\}_{p,q=1}^{P,Q}$ where $\mathbf{v}_{p,q} \in \mathbb{R}^{M_{p,q}}$ at locations $\mathbf{Z}^{(w)} = \{\mathbf{Z}_{p,q}\}_{p,q=1}^{P,Q}$ for $\mathbf{Z}_{p,q} \in \mathbb{R}^{M_{p,q}, D}$. We assume that \mathbf{f} and \mathbf{W} are independent Gaussian processes, and furthermore that they factor across components. We then write the joint probability density including the inducing points as

$$\begin{aligned} p(\mathbf{Y}, \mathbf{W}, \mathbf{f}, \mathbf{u}, \mathbf{v}) &\propto p(\mathbf{Y}|\mathbf{W}, \mathbf{f}) p(\mathbf{W}, \mathbf{f}|\mathbf{u}, \mathbf{v}) p(\mathbf{u}, \mathbf{v}) \\ &= p(\mathbf{Y}|\mathbf{W}, \mathbf{f}) p(\mathbf{f}|\mathbf{u}) p(\mathbf{W}|\mathbf{v}) p(\mathbf{u}) p(\mathbf{v}) \\ &= p(\mathbf{Y}|\mathbf{W}, \mathbf{f}) \prod_{p=1}^P \prod_{q=1}^Q p(\mathbf{W}_{p,q}|\mathbf{v}_{p,q}) p(\mathbf{v}_{p,q}) \prod_{q=1}^Q p(\mathbf{f}_q|\mathbf{u}_q) p(\mathbf{u}_q) \end{aligned} \quad (14)$$

where $p(\mathbf{f}_q|\mathbf{u}_q) = \mathcal{N}(\mathbf{f}_q|\boldsymbol{\mu}_q^{(u)}, \boldsymbol{\Sigma}_q^{(u)})$ and $p(\mathbf{W}_{p,q}|\mathbf{v}_{p,q}) = \mathcal{N}(\mathbf{W}_{p,q}|\boldsymbol{\mu}_{p,q}^{(v)}, \boldsymbol{\Sigma}_{p,q}^{(v)})$ have the standard conditional Gaussian distributions with parameters

$$\begin{aligned} \boldsymbol{\mu}_q^{(u)} &= \mathbf{K}_q^{(u)}(\mathbf{X}, \mathbf{Z}_q) \mathbf{K}_q^{(u)}(\mathbf{Z}_q, \mathbf{Z}_q)^{-1} \mathbf{u}_q \\ \boldsymbol{\mu}_{p,q}^{(v)} &= \mathbf{K}_{p,q}^{(v)}(\mathbf{X}, \mathbf{Z}_{p,q}) \mathbf{K}_{p,q}^{(v)}(\mathbf{Z}_{p,q}, \mathbf{Z}_{p,q})^{-1} \mathbf{v}_{p,q} \\ \boldsymbol{\Sigma}_{\cdot,\cdot}^{(\cdot)} &= \mathbf{K}_{\cdot,\cdot}^{(\cdot)}(\mathbf{X}, \mathbf{X}) - \mathbf{K}_{\cdot,\cdot}^{(\cdot)}(\mathbf{X}, \mathbf{Z}_{\cdot,\cdot}) \mathbf{K}_{\cdot,\cdot}^{(\cdot)}(\mathbf{Z}_{\cdot,\cdot}, \mathbf{Z}_{\cdot,\cdot})^{-1} \mathbf{K}_{\cdot,\cdot}^{(\cdot)}(\mathbf{Z}_{\cdot,\cdot}, \mathbf{X}) \end{aligned} \quad (15)$$

and $p(\mathbf{u}_q) = \mathcal{N}(\mathbf{u}_q|0, \mathbf{K}_q^{(u)})$ and $p(\mathbf{v}_{p,q}) = \mathcal{N}(\mathbf{v}_{p,q}|0, \mathbf{K}_{p,q}^{(v)})$.

A.1 Approximate Posterior

Variational inference turns the Bayesian posterior inference problem into an optimisation problem where the objective function is the evidence lower bound (ELBO). Following [10] we construct our variational lower bound, keeping \mathbf{u} and \mathbf{v} explicit, allowing for stochastic variational inference. We define our approximate posterior to have the factorisation

$$q(\mathbf{u}, \mathbf{v}, \mathbf{f}, \mathbf{W}) = p(\mathbf{f}, \mathbf{W}|\mathbf{u}, \mathbf{v}) q(\mathbf{u}, \mathbf{v}) \quad (16)$$

where we have defined $q(\mathbf{u}, \mathbf{v})$ to be a free-form mixture of Gaussians

$$q(\mathbf{u}, \mathbf{v}) = \sum_{k=1}^K \pi_k \prod_{j=1}^Q \mathcal{N}(\mathbf{u}_j | \mathbf{m}_{k,j}^{(\mathbf{u})}, \mathbf{S}_{k,j}^{(\mathbf{u})}) \cdot \prod_{i,j=1}^{P,Q} \mathcal{N}(\mathbf{v}_{i,j} | \mathbf{m}_{k,i,j}^{(\mathbf{v})}, \mathbf{S}_{k,i,j}^{(\mathbf{v})}) \quad (17)$$

where each mean $\mathbf{m}_k \in \mathbb{R}^M$ and covariance factor $\mathbf{S}_k \in \mathbb{R}^{M \times M}$. The marginal $q(\mathbf{f}, \mathbf{W})$ is then derived as

$$\begin{aligned}
q(\mathbf{f}, \mathbf{W}) &= \int p(\mathbf{f}, \mathbf{W} | \mathbf{u}, \mathbf{v}) q(\mathbf{u}, \mathbf{v}) d\mathbf{u} d\mathbf{v} \\
&= \sum_{k=1}^K \pi_k \int \prod_{q=1}^Q p(\mathbf{f}_q | \mathbf{u}_{k,q}) q(\mathbf{u}_{k,q}) d\mathbf{u}_{k,q} \cdot \int \prod_{p,q=1}^{P,Q} p(\mathbf{W}_{p,q} | \mathbf{v}_{k,p,q}) q(\mathbf{v}_{k,p,q}) d\mathbf{v}_{k,p,q} \\
&= \sum_{k=1}^K \pi_k \prod_{q=1}^Q \mathcal{N}(\mathbf{f}_q | \boldsymbol{\mu}_{k,q}^{(f)}, \boldsymbol{\Sigma}_{k,q}^{(f)}) \cdot \prod_{p,q=1}^{P,Q} \mathcal{N}(\mathbf{W}_{p,q} | \boldsymbol{\mu}_{k,p,q}^{(W)}, \boldsymbol{\Sigma}_{k,p,q}^{(W)})
\end{aligned} \tag{18}$$

where the moments of the variational distribution are given by

$$\begin{aligned}
\boldsymbol{\mu}_{k,\cdot,\cdot}^{(\cdot)} &= \mathbf{K}^{(\cdot)}(\mathbf{X}, \mathbf{Z}) \mathbf{K}^{(\cdot)}(\mathbf{Z}, \mathbf{Z})^{-1} \mathbf{m}_{k,\cdot}^{(\cdot)} \\
\boldsymbol{\Sigma}_{k,\cdot,\cdot}^{(\cdot)} &= \boldsymbol{\Sigma}_{k,\cdot,\cdot}'^{(\cdot)} + \mathbf{K}^{(\cdot)}(\mathbf{X}, \mathbf{Z}) \mathbf{K}^{(\cdot)}(\mathbf{Z}, \mathbf{Z})^{-1} \mathbf{S}_{k,\cdot,\cdot}^{(\cdot)} \mathbf{K}^{(\cdot)}(\mathbf{Z}, \mathbf{Z})^{-1} \mathbf{K}^{(\cdot)}(\mathbf{Z}, \mathbf{X})
\end{aligned} \tag{19}$$

A.2 Variational Lower Bound

Following [14] we derive the ELBO as:

$$\begin{aligned}
\mathcal{L}_{\text{MR-GPRN}} &= \mathbb{E}_q \left[\log \frac{p(\mathbf{Y}, \mathbf{W}, \mathbf{f}, \mathbf{u}, \mathbf{v})}{q(\mathbf{W}, \mathbf{f}, \mathbf{u}, \mathbf{v})} \right] \\
&= \mathbb{E}_q \left[\log \frac{p(\mathbf{Y} | \mathbf{W}, \mathbf{f}) p(\mathbf{f} | \mathbf{u}) p(\mathbf{u}) p(\mathbf{W} | \mathbf{v}) p(\mathbf{v})}{p(\mathbf{f} | \mathbf{u}) p(\mathbf{W} | \mathbf{v}) q(\mathbf{u}, \mathbf{v})} \right] \\
&= \underbrace{\mathbb{E}_{q(\mathbf{f}, \mathbf{W})} [\log p(\mathbf{Y} | \mathbf{f}, \mathbf{W})]}_{\text{ELL}} + \underbrace{\mathbb{E}_{q(\mathbf{u}, \mathbf{v})} \left[\log \frac{p(\mathbf{u}, \mathbf{v})}{q(\mathbf{u}, \mathbf{v})} \right]}_{\text{KL}}
\end{aligned} \tag{20}$$

The subsequent sections derive the closed-form expressions of both the expected log likelihood (ELL) and the KL term.

A.3 MR-GPRN: KL Term

Following [14] the KL term is decomposed into two terms:

$$\text{KL} = \mathbb{E}_q \left[\log \frac{p(\mathbf{u}, \mathbf{v})}{q(\mathbf{u}, \mathbf{v})} \right] = \underbrace{\mathbb{E}_q [\log p(\mathbf{u}, \mathbf{v})]}_{\text{cross}} - \underbrace{\mathbb{E}_q [\log q(\mathbf{u}, \mathbf{v})]}_{\text{ent}} \tag{21}$$

where we deal with each term separately.

A.3.1 Cross Term

The cross term is calculated as

$$\begin{aligned}
\text{cross} &= \mathbb{E}_q [\log p(\mathbf{u}, \mathbf{v})] = \sum_{k=1}^K \pi_k \mathbb{E}_{q_k(\mathbf{u}, \mathbf{v})} [\log p(\mathbf{u}, \mathbf{v})] \\
&= \sum_{k=1}^K \pi_k \left(\sum_{q=1}^Q \mathbb{E}_{q(\mathbf{u}_{k,q})} [\log p(\mathbf{u}_q)] + \sum_{p=1}^P \sum_{q=1}^Q \mathbb{E}_{q(\mathbf{v}_{k,p,q})} [\log p(\mathbf{v}_{p,q})] \right)
\end{aligned} \tag{22}$$

where each

$$\begin{aligned}
\mathbb{E}_{q(\mathbf{u}_{k,q})} [\log p(\mathbf{u}_q)] &= C_u - \frac{1}{2} (\mathbf{m}_{k,q}^{(m)})^T \mathbf{K}_q^u(\mathbf{Z}_q^{(u)}, \mathbf{Z}_q^{(u)})^{-1} \mathbf{m}_{k,q}^{(u)} - \frac{1}{2} \text{Tr} \left[\mathbf{K}_q^u(\mathbf{Z}_q^{(u)}, \mathbf{Z}_q^{(u)})^{-1} \mathbf{S}_{k,q}^u \right] \\
\mathbb{E}_{q(\mathbf{v}_{k,p,q})} [\log p(\mathbf{v}_{p,q})] &= C_v - \frac{1}{2} (\mathbf{m}_{k,p,q}^{(v)})^T \mathbf{K}_{p,q}^v(\mathbf{Z}_{p,q}^{(v)}, \mathbf{Z}_{p,q}^{(v)})^{-1} \mathbf{m}_{k,p,q}^{(v)} - \frac{1}{2} \text{Tr} \left[\mathbf{K}_{p,q}^v(\mathbf{Z}_{p,q}^{(v)}, \mathbf{Z}_{p,q}^{(v)})^{-1} \mathbf{S}_{k,p,q}^v \right]
\end{aligned} \tag{23}$$

and

$$\begin{aligned} C_u &= -\frac{M_q}{2} \log(2\pi) - \frac{1}{2} \log |\mathbf{K}_q^{(u)}(\mathbf{Z}_q^{(u)}, \mathbf{Z}_q^{(u)})| \\ C_v &= -\frac{M_{p,q}}{2} \log(2\pi) - \frac{1}{2} \log |\mathbf{K}_{p,q}^{(v)}(\mathbf{Z}_{p,q}^{(v)}, \mathbf{Z}_{p,q}^{(v)})| \end{aligned} \quad (24)$$

A.3.2 Entropy Term

Following [14] we lower bound the entropy term of the mixture of Gaussians as

$$\begin{aligned} \text{ent} &= -\sum_{k=1}^K \pi_k \mathbb{E}_{q_k(\mathbf{u}, \mathbf{v})} [\log q(\mathbf{u}, \mathbf{v})] \\ &\geq -\sum_{k=1}^K \pi_k \log (\mathbb{E}_{q_k(\mathbf{u}, \mathbf{v})} [q(\mathbf{u}, \mathbf{v})]) \\ &= -\sum_{k=1}^K \pi_k \log \left(\sum_{l=1}^K \pi_l \mathbb{E}_{q_k(\mathbf{u})} [q_l(\mathbf{u})] \cdot \mathbb{E}_{q_k(\mathbf{v})} [q_l(\mathbf{v})] \right) \\ &= -\sum_{k=1}^K \pi_k \log \left(\sum_{l=1}^K \pi_l \mathcal{N}(\mathbf{m}_k^{(u)} | \mathbf{m}_l^{(u)}, \mathbf{S}_k^{(u)} + \mathbf{S}_l^{(u)}) \cdot \mathcal{N}(\mathbf{m}_k^{(v)} | \mathbf{m}_l^{(v)}, \mathbf{S}_k^{(v)} + \mathbf{S}_l^{(v)}) \right). \end{aligned} \quad (25)$$

A.4 MR-GPRN: Closed Form Expected Log Likelihood

We now derive the closed form expected log likelihood (ELL) in Eq. 20. The ELL is

$$\text{ELL} = \underbrace{\sum_{a=1}^A \sum_{p=1}^P \sum_{n=1}^{N_a} \sum_{k=1}^K \pi_k \mathbb{E}_{q_k(\mathbf{f}, \mathbf{W})} \left[\log \mathcal{N}(\mathbf{Y}_{a,p,n} | \frac{1}{|\mathcal{S}_{a,n}|} \sum_{\mathbf{x} \in \mathcal{S}_{a,n}} \sum_{q=1}^Q \mathbf{W}_{p,q}(\mathbf{x}) \mathbf{f}_q(\mathbf{x}), \sigma_a^2) \right]}_{\text{ELL}_a} \quad (26)$$

where each of the components can now be dealt with separately. Dealing with component a

$$\begin{aligned} \text{ELL}_a &= \mathbb{E}_{q_k(\mathbf{f}, \mathbf{W})} \left[\log \mathcal{N}(\mathbf{Y}_{a,p,n} | \frac{1}{|\mathcal{S}_{a,n}|} \sum_{\mathbf{x} \in \mathcal{S}_{a,n}} \sum_{q=1}^Q \mathbf{W}_{p,q}(\mathbf{x}) \mathbf{f}_q(\mathbf{x}), \sigma_a^2) \right] \\ &= C_1 + C_2 \mathbb{E}_{q_k(\mathbf{f}, \mathbf{W})} [(\mathbf{Y}_{a,p,n} - \mu_y)^T (\mathbf{Y}_{a,p,n} - \mu_y)] \\ &= C_1 + C_2 (\mathbb{E}_{q_k(\mathbf{f}, \mathbf{W})} [\mathbf{Y}_{a,p,n}^T \mathbf{Y}_{a,p,n}] - 2 \cdot \mathbb{E}_{q_k(\mathbf{f}, \mathbf{W})} [\mathbf{Y}_{a,p,n} \mu_y] + \mathbb{E}_{q_k(\mathbf{f}, \mathbf{W})} [\mu_y^2]) \end{aligned} \quad (27)$$

where

$$\begin{aligned} \mu_y &= \frac{1}{|\mathcal{S}_{a,n}|} \sum_{\mathbf{x} \in \mathcal{S}_{a,n}} \sum_{q=1}^Q \mathbf{W}_{p,q}(\mathbf{x}) \mathbf{f}_q(\mathbf{x}) \\ C_1 &= -\frac{N_a}{2} \log(2\pi \sigma_{a,p}^2) \\ C_2 &= -\frac{1}{2\sigma_{a,p}^2} \end{aligned} \quad (28)$$

and we now deal with each of these expectations separately.

A.4.1 ELL: 1st Term

The first expectation does not contain \mathbf{f} or \mathbf{W} and so the expectations can be dropped

$$\mathbb{E}_{q_k(\mathbf{f}, \mathbf{W})} [\mathbf{Y}_{a,p,n}^2] = \mathbf{Y}_{a,p,n}^2 \quad (29)$$

A.4.2 ELL: 2nd Term

For the second term the expectation is brought inside the sum, and then applied to each of \mathbf{f} and \mathbf{W} separately

$$\begin{aligned}\mathbb{E}_{q_k(\mathbf{f}, \mathbf{W})} [\mu_y \mathbf{Y}_{a,p,n}] &= \left(\frac{1}{|\mathcal{S}_{a,n}|} \sum_{\mathbf{x} \in \mathcal{S}_{a,n}} \sum_{q=1}^Q \mathbb{E}_q [\mathbf{W}_{p,q}(\mathbf{x})] \mathbb{E}_q [\mathbf{f}_q(\mathbf{x})] \right) \mathbf{Y}_{a,p,n} \\ &= \left(\frac{1}{|\mathcal{S}_{a,n}|} \sum_{\mathbf{x} \in \mathcal{S}_{a,n}} \sum_{q=1}^Q \boldsymbol{\mu}_{k,p,q}^{(w)}(\mathbf{x}) \boldsymbol{\mu}_{k,q}^{(f)}(\mathbf{x}) \right) \mathbf{Y}_{a,p,n}\end{aligned}\quad (30)$$

where we have used the independence property, $q(\mathbf{f}, \mathbf{W}) = q(\mathbf{f})q(\mathbf{W})$.

A.4.3 ELL: 3rd Term

In the last expectation we have a product of sums, that is then expanded into a quadruple sum over q_1, q_2 . There will be two cases; the first is when $q_1 = q_2$ inside the sum \mathbf{f} and \mathbf{W} will appear in square forms, and the second when the expectations can be treated as in Sec. A.4.2.

The final term obtained after expanding the quadratic in (27) is given by

$$\mathbb{E}_{q_k(\mathbf{f}, \mathbf{W})} [\mu_y^2] = \frac{1}{|\mathcal{S}_{a,n}|^2} \sum_{q_1=1}^Q \sum_{q_2=1}^Q \sum_{\mathbf{x}_1} \sum_{\mathbf{x}_2} \mathbb{E}_{q_k(\mathbf{f}, \mathbf{W})} [\mathbf{f}_{q_1}(\mathbf{x}_1) \mathbf{W}_{p,q_1}(\mathbf{x}_1) \mathbf{W}_{p,q_2}(\mathbf{x}_2) \mathbf{f}_{q_2}(\mathbf{x}_2)] \quad (31)$$

For the case where $q_1 = q_2$, then upon taking expectations we have

$$\begin{aligned}\mathbb{E}_{q_k(\mathbf{f}, \mathbf{W})} [(\mathbf{f}_{q_1}(\mathbf{x}_1) \mathbf{W}_{p,q_1}(\mathbf{x}_1) \mathbf{W}_{p,q_1}(\mathbf{x}_2) \mathbf{f}_{q_1}(\mathbf{x}_2))] \\ = \mathbb{E}_{q_k(\mathbf{f})} \left[\mathbf{f}_{q_1}(\mathbf{x}_1) \left(\boldsymbol{\mu}_{k,p,q_1}^{(w)}(\mathbf{x}_1) \boldsymbol{\mu}_{k,p,q_1}^{(w)}(\mathbf{x}_2) + \boldsymbol{\Sigma}_{k,p,q_1}^{(w)}(\mathbf{x}_1, \mathbf{x}_2) \right) \mathbf{f}_{q_1}(\mathbf{x}_2) \right] \\ = \boldsymbol{\Sigma}_{k,p,q_1}^{(w)}(\mathbf{x}_1, \mathbf{x}_2) \boldsymbol{\Sigma}_{k,q_1}^{(f)}(\mathbf{x}_1, \mathbf{x}_2) + \boldsymbol{\mu}_{k,q_1}^{(f)}(\mathbf{x}_1) \boldsymbol{\Sigma}_{k,p,q_1}^{(w)}(\mathbf{x}_1, \mathbf{x}_2) \boldsymbol{\mu}_{k,q_1}^{(f)}(\mathbf{x}_2) + \\ \boldsymbol{\mu}_{k,p,q_1}^{(w)}(\mathbf{x}_1) \boldsymbol{\Sigma}_{k,q_1}^{(f)}(\mathbf{x}_1, \mathbf{x}_2) \boldsymbol{\mu}_{k,p,q_1}^{(w)}(\mathbf{x}_2) + \boldsymbol{\mu}_{k,q_1}^{(f)}(\mathbf{x}_1) \boldsymbol{\mu}_{k,p,q_1}^{(w)}(\mathbf{x}_1) \boldsymbol{\mu}_{k,p,q_1}^{(w)}(\mathbf{x}_2) \boldsymbol{\mu}_{k,q_1}^{(f)}(\mathbf{x}_2)\end{aligned}\quad (32)$$

where we use the notation $\boldsymbol{\Sigma}_{\cdot,\cdot}^{(\cdot)}(\mathbf{x}_1, \mathbf{x}_2)$ to denote entry of the covariance matrix which agrees with the enumeration of \mathbf{x}_1 and \mathbf{x}_2 . And similarly $\boldsymbol{\mu}_{k,p,q_1}^{(w)}(\mathbf{x}_1), \boldsymbol{\mu}_{k,\cdot}^{(\cdot)}(\mathbf{x}_1)$ denotes the entry of the mean vector agreeing with the enumeration of \mathbf{x}_1 .

When $q_1 \neq q_2$ there will be no square terms and so the expectation will simply be

$$\mathbb{E}_{q_k(\mathbf{f}, \mathbf{W})} [\mathbf{f}_{q_1}(\mathbf{x}_1) \mathbf{W}_{p,q_1}(\mathbf{x}_1) \mathbf{W}_{p,q_2}(\mathbf{x}_2) \mathbf{f}_{q_2}(\mathbf{x}_2)] = \boldsymbol{\mu}_{k,q_1}^{(f)}(\mathbf{x}_1) \boldsymbol{\mu}_{k,p,q_1}^{(w)}(\mathbf{x}_1) \boldsymbol{\mu}_{k,p,q_1}^{(w)}(\mathbf{x}_2) \boldsymbol{\mu}_{k,q_1}^{(f)}(\mathbf{x}_2) \quad (33)$$

The complete third term can now be rewritten as

$$\begin{aligned}\mathbb{E}_q [\mu_y^2] &= \frac{1}{|\mathcal{S}_{a,n}|^2} \sum_{q_1=1}^Q \sum_{q_2=1}^Q \sum_{\mathbf{x}_1} \sum_{\mathbf{x}_2} \boldsymbol{\mu}_{k,q_1}^{(f)}(\mathbf{x}_1) \boldsymbol{\mu}_{k,p,q_1}^{(w)}(\mathbf{x}_1) \boldsymbol{\mu}_{k,p,q_1}^{(w)}(\mathbf{x}_2) \boldsymbol{\mu}_{k,q_1}^{(f)}(\mathbf{x}_2) + \\ &\quad \frac{1}{|\mathcal{S}_{a,n}|^2} \sum_{q=1}^Q \sum_{\mathbf{x}_1} \sum_{\mathbf{x}_2} \boldsymbol{\Sigma}_{k,p,q}^{(w)}(\mathbf{x}_1, \mathbf{x}_2) \boldsymbol{\Sigma}_{k,q}^{(f)}(\mathbf{x}_1, \mathbf{x}_2) \\ &\quad + \boldsymbol{\mu}_{k,q}^{(f)}(\mathbf{x}_1) \boldsymbol{\Sigma}_{k,p,q}^{(w)}(\mathbf{x}_1, \mathbf{x}_2) \boldsymbol{\mu}_{k,q}^{(f)}(\mathbf{x}_2) + \boldsymbol{\mu}_{k,p,q}^{(w)}(\mathbf{x}_1) \boldsymbol{\Sigma}_{k,q}^{(f)}(\mathbf{x}_1, \mathbf{x}_2) \boldsymbol{\mu}_{k,p,q}^{(w)}(\mathbf{x}_2)\end{aligned}\quad (34)$$

A.4.4 Full Term

Combining the derivations above we have

$$\begin{aligned} \text{ELL} = & \sum_{a=1}^{\mathcal{A}} \sum_{k=1}^K \pi_k \frac{1}{2\sigma_a^2} \left[\mathbf{Y}_{a,p,n}^2 - 2 \left(\frac{1}{|\mathcal{S}_{a,n}|} \sum_{\mathbf{x} \in \mathcal{S}_{a,n}} \sum_{q=1}^Q \boldsymbol{\mu}_{k,p,q}^{(w)}(\mathbf{x}) \boldsymbol{\mu}_{k,q}^{(f)}(\mathbf{x}) \right) \mathbf{Y}_{a,p,n} \right. \\ & + \frac{1}{|\mathcal{S}_{a,n}|^2} \left(\sum_{q_1, q_2=1}^Q \sum_{\mathbf{x}_1, \mathbf{x}_2}^{\mathcal{S}_{a,n}} \boldsymbol{\mu}_{k,q_1}^{(f)}(\mathbf{x}_1) \boldsymbol{\mu}_{k,p,q_1}^{(w)}(\mathbf{x}_1) \boldsymbol{\mu}_{k,p,q_1}^{(w)}(\mathbf{x}_2) \boldsymbol{\mu}_{k,q_1}^{(f)}(\mathbf{x}_2) \right. \\ & \left. \left. + \delta_{q_1, q_2} \cdot \left(\boldsymbol{\Sigma}_{k,p,q}^{(w)}(\mathbf{x}_1, \mathbf{x}_2) \boldsymbol{\Sigma}_{k,q}^{(f)}(\mathbf{x}_1, \mathbf{x}_2) + \boldsymbol{\mu}_{k,q}^{(f)}(\mathbf{x}_1) \boldsymbol{\Sigma}_{k,p,q}^{(w)}(\mathbf{x}_1, \mathbf{x}_2) \boldsymbol{\mu}_{k,q}^{(f)}(\mathbf{x}_2) \boldsymbol{\mu}_{k,p,q}^{(w)}(\mathbf{x}_1) \boldsymbol{\Sigma}_{k,q}^{(f)}(\mathbf{x}_1, \mathbf{x}_2) \boldsymbol{\mu}_{k,p,q}^{(w)}(\mathbf{x}_2) \right) \right] \right. \\ & \left. - \frac{1}{2} \sum_{a=1}^{\mathcal{A}} N_a P \log(2\pi\sigma_a^2) \right] \end{aligned}$$

or

$$\begin{aligned} \text{ELL} = & \sum_{a=1}^{\mathcal{A}} \sum_{p=1}^P \sum_{n=1}^{N_a} \sum_{k=1}^K \pi_k \log \mathcal{N} \left(\mathbf{Y}_{a,p,n} \mid \frac{1}{|\mathcal{S}_{a,n}|} \sum_{\mathbf{x} \in \mathcal{S}_{a,n}} \sum_{q=1}^Q \boldsymbol{\mu}_{k,p,q}^{(w)}(\mathbf{x}) \boldsymbol{\mu}_{k,q}^{(f)}(\mathbf{x}), \sigma_{a,p}^2 \right) \\ & - \sum_{a=1}^{\mathcal{A}} \sum_{p=1}^P \sum_{n=1}^{N_a} \sum_{k=1}^K \frac{\pi_k}{2\sigma_{a,p}^2} \frac{1}{|\mathcal{S}_{a,n}|^2} \sum_{q=1}^Q \sum_{\mathbf{x}_1, \mathbf{x}_2} \boldsymbol{\Sigma}_{k,p,q}^{(w)}(\mathbf{x}_1, \mathbf{x}_2) \boldsymbol{\Sigma}_{k,q}^{(f)}(\mathbf{x}_1, \mathbf{x}_2) \\ & + \boldsymbol{\mu}_{k,q}^{(f)}(\mathbf{x}_1) \boldsymbol{\Sigma}_{k,p,q}^{(w)}(\mathbf{x}_1, \mathbf{x}_2) \boldsymbol{\mu}_{k,q}^{(f)}(\mathbf{x}_2) \boldsymbol{\mu}_{k,p,q}^{(w)}(\mathbf{x}_1) \boldsymbol{\Sigma}_{k,q}^{(f)}(\mathbf{x}_1, \mathbf{x}_2) \boldsymbol{\mu}_{k,p,q}^{(w)}(\mathbf{x}_2) \end{aligned} \quad (35)$$

A.5 MR-GPRN: Closed Form Expected Log Likelihood ($\mathbf{W} \rightarrow \exp(\mathbf{W})$)

A.5.1 Moments of exponentiated Gaussian random variables

In this section we provide additional results needed for the calculations in this section using standard properties of the moment generating function of a Gaussian random variable.

$$\begin{aligned} \mathbb{E}[\exp(t \cdot \mathbf{W}_{p,q})] &= \int_{\mathbb{R}} \mathcal{N}(\mathbf{W}_{p,q} \mid \mu_{\mathbf{W}_{p,q}}, \Sigma_{\mathbf{W}_{p,q}}) \exp(t \cdot \mathbf{W}_{p,q}) d\mathbf{W}_{p,q} \\ &= \exp(t\mu_{\mathbf{W}_{p,q}} + \frac{t^2}{2} \Sigma_{\mathbf{W}_{p,q}}) \end{aligned} \quad (36)$$

where $\exp(\cdot)$ is defined as element-wise function, for arbitrary $t \in \mathbb{R}$.

A.5.2 ELL with positive weights

If \mathbf{W} is passed through an exponential function to enforce positive latent weights the expected log-likelihood is

$$\alpha \sum_{a=1}^{\mathcal{A}} \sum_{p=1}^P \sum_{n=1}^{N_a} \sum_{k=1}^K \pi_k \underbrace{\mathbb{E}_{q_k(\mathbf{f}, \mathbf{W})} \left[\log \mathcal{N}(\mathbf{Y}_{a,p,n} \mid \frac{1}{|\mathcal{S}_{a,n}|} \sum_{\mathbf{x} \in \mathcal{S}_{a,n}} \sum_{q=1}^Q \exp(\mathbf{W}_{p,q}(\mathbf{x})) \mathbf{f}_q(\mathbf{x}), \sigma_a^2) \right]}_{\text{ELL}_a} \quad (37)$$

Each of the likelihood components can now be dealt with separately, and moreover has the same form as Eq. 27 with

$$\mu_y = \frac{1}{|\mathcal{S}_{a,n}|} \sum_{\mathbf{x} \in \mathcal{S}_{a,n}} \sum_{q=1}^Q \exp(\mathbf{W}_{p,q}(\mathbf{x})) \mathbf{f}_q(\mathbf{x}) \quad (38)$$

As above, we now deal with the expectation of each term obtained after expanding the quadratic separately, but using the results for the exponentiated moments in A.5.1.

A.5.3 ELL: 1st Term

As in Sec. A.4.1 the expectation is constant:

$$\mathbb{E}_{q_k(\mathbf{f}, \mathbf{W})} [\mathbf{Y}_{a,p,n}^2] = \mathbf{Y}_{a,p,n}^2 \quad (39)$$

A.5.4 ELL: 2nd Term

As in Sec. A.4.2 the 2nd term the expectation is brought inside the sum and applied to \mathbf{f} and \mathbf{W} separately. To evaluate the expectation of $\exp(\mathbf{W})$ we use the result from Eq. 36:

$$\begin{aligned} \mathbb{E}_{q_k(\mathbf{f}, \mathbf{W})} [\mu_y \mathbf{Y}_{a,p,n}] &= \left(\frac{1}{|\mathcal{S}_{a,n}|} \sum_{\mathbf{x} \in \mathcal{S}_{a,n}} \sum_{q=1}^Q \mathbb{E}_{q_k(\mathbf{f}, \mathbf{W})} [\exp(\mathbf{W}_{p,q}(\mathbf{x}))] \mathbb{E}_{q_k(\mathbf{f}, \mathbf{W})} [\mathbf{f}_q(\mathbf{x})] \right) \mathbf{Y}_{a,p,n} \\ &= \left(\frac{1}{|\mathcal{S}_{a,n}|} \sum_{\mathbf{x} \in \mathcal{S}_{a,n}} \sum_{q=1}^Q \exp \left(\mu_{k,p,q}^{(w)}(\mathbf{x}) + \frac{1}{2} \Sigma_{k,p,q}^{(w)}(\mathbf{x}, \mathbf{x}) \right) \mu_{k,q}^{(f)}(\mathbf{x}) \right) \mathbf{Y}_{a,p,n}. \end{aligned} \quad (40)$$

A.5.5 ELL: 3rd Term

This derivation closely follows that from Section A.4.3. The third term is given by

$$\mathbb{E}_{q_k(\mathbf{f}, \mathbf{W})} [\mu_y^2] = \frac{1}{|\mathcal{S}_{a,n}|^2} \sum_{q_1=1}^Q \sum_{q_2=1}^Q \sum_{\mathbf{x}_1} \sum_{\mathbf{x}_2} \mathbb{E}_{q_k(\mathbf{f}, \mathbf{W})} [\mathbf{f}_{q_1}(\mathbf{x}_1) \exp(\mathbf{W}_{p,q_1}(\mathbf{x}_1)) \exp(\mathbf{W}_{p,q_2}(\mathbf{x}_2)) \mathbf{f}_{q_2}(\mathbf{x}_2)] \quad (41)$$

For the case where $q_1 = q_2$

$$\begin{aligned} &\mathbb{E}_{q_k(\mathbf{f}, \mathbf{W})} [\mathbf{f}_{q_1}(\mathbf{x}_1) \exp(\mathbf{W}_{p,q_1}(\mathbf{x}_1)) \exp(\mathbf{W}_{p,q_1}(\mathbf{x}_2)) \mathbf{f}_{q_1}(\mathbf{x}_2)] \\ &= \mathbb{E}_{q_k(\mathbf{f})} \left[\mathbf{f}_{q_1}(\mathbf{x}_1) \left(\tilde{\mu}_{k,p,q_1}^{(w)}(\mathbf{x}_1) \tilde{\mu}_{k,p,q_1}^{(w)}(\mathbf{x}_2) + \tilde{\Sigma}_{k,p,q}^{(w)}(\mathbf{x}_1, \mathbf{x}_2) \right) \mathbf{f}_{q_1}(\mathbf{x}_2) \right] \\ &= \tilde{\Sigma}_{k,p,q_1}^{(w)}(\mathbf{x}_1, \mathbf{x}_2) \Sigma_{k,q_1}^{(f)}(\mathbf{x}_1, \mathbf{x}_2) + \mu_{k,q_1}^{(f)}(\mathbf{x}_1) \tilde{\Sigma}_{k,p,q_1}^{(w)}(\mathbf{x}_1, \mathbf{x}_2) \mu_{k,q_1}^{(f)}(\mathbf{x}_2) + \\ &\quad \tilde{\mu}_{k,p,q_1}^{(w)}(\mathbf{x}_1) \Sigma_{k,q_1}^{(f)}(\mathbf{x}_1, \mathbf{x}_2) \tilde{\mu}_{k,p,q_1}^{(w)}(\mathbf{x}_2) + \mu_{k,q_1}^{(f)}(\mathbf{x}_1) \tilde{\mu}_{k,p,q_1}^{(w)}(\mathbf{x}_1) \tilde{\mu}_{k,p,q_1}^{(w)}(\mathbf{x}_2) \mu_{k,q_1}^{(f)}(\mathbf{x}_2) \end{aligned} \quad (42)$$

where

$$\tilde{\mu}_{k,p,q}^{(w)}(\mathbf{x}_1) = \exp(\mu_{k,p,q}^{(w)}(\mathbf{x}_1) + \frac{1}{2} \Sigma_{k,p,q}^{(w)}(\mathbf{x}_1, \mathbf{x}_1)),$$

and

$$\tilde{\Sigma}_{k,p,q}^{(w)}(\mathbf{x}_1, \mathbf{x}_2) = \exp(2(\mu_{k,p,q}^{(w)} + \Sigma_{k,p,q}^{(w)}(\mathbf{x}_1, \mathbf{x}_2))).$$

In all other cases there will be no square terms and so the expectation will simply be

$$\begin{aligned} &\mathbb{E}_{q_k(\mathbf{f}, \mathbf{W})} [\mathbf{f}_{q_1}(\mathbf{x}_1) \exp(\mathbf{W}_{p,q_1}(\mathbf{x}_1)) \exp(\mathbf{W}_{p,q_2}(\mathbf{x}_2)) \mathbf{f}_{q_2}(\mathbf{x}_2)] \\ &= \mu_{k,q_1}^{(f)}(\mathbf{x}_1) \tilde{\mu}_{k,p,q_1}^{(w)}(\mathbf{x}_1) \tilde{\mu}_{k,p,q_2}^{(w)}(\mathbf{x}_2) \mu_{k,q_2}^{(f)}(\mathbf{x}_2) \end{aligned} \quad (43)$$

A.6 Prediction

Although the full predictive distribution for MR-GPRN is not available analytically, we are able to derive the first and second moments in closed form.

A.6.1 Predictive Mean

To calculate the first moment we use the approximate variational posterior in place of the true posterior. The predictive mean for task p is given as

$$\begin{aligned}
\mathbb{E}[\mathbf{y}_p^* \mid \mathbf{x}^*, \mathbf{X}, \mathbf{Y}] &= \int \mathbf{y}_p^* p(\mathbf{y}_p^* \mid \mathbf{X}^*, \mathbf{X}, \mathbf{Y}) d\mathbf{y}^* \\
&\approx \sum_{k=1}^K \pi_k \int \mathbf{y}_p^* p(\mathbf{y}_p^* \mid \mathbf{W}_{k,p}^*, \mathbf{f}_k^*, \mathbf{x}^*, \mathbf{X}, \mathbf{Y}) q(\mathbf{f}_k^*) q(\mathbf{W}_k^*) d\mathbf{y}_p^* d\mathbf{W}_{k,p}^* d\mathbf{f}_k^* \\
&= \sum_{k=1}^K \pi_k \int \sum_{q=1}^Q \mathbf{W}_{p,q}^* \mathbf{f}_q^* q(\mathbf{f}_k^*) q(\mathbf{W}_k^*) d\mathbf{W}_{k,p}^* d\mathbf{f}_k^* \\
&= \sum_{k=1}^K \pi_k \sum_{q=1}^Q \boldsymbol{\mu}_{k,p,q}^{(w)}(\mathbf{x}^*) \boldsymbol{\mu}_{k,q}^{(f)}(\mathbf{x}^*)
\end{aligned} \tag{44}$$

A.6.2 Predictive Variance

The point wise second moment of task p is given by

$$\mathbb{V}[\mathbf{y}_p^*] = \mathbb{E}[(\mathbf{y}_p^*)^2] - \mathbb{E}[\mathbf{y}_p^*] \mathbb{E}[\mathbf{y}_p^*] \tag{45}$$

We have already calculated the closed form mean in Eq. 44 and the square form is given by

$$\begin{aligned}
\mathbb{E}[(\mathbf{y}_p^*)^2] &\approx \sum_{k=1}^K \pi_k \int (\mathbf{y}_p^*)^2 p(\mathbf{y}_p^* \mid \mathbf{W}_{k,p}^*, \mathbf{f}_k^*, \mathbf{x}^*, \mathbf{X}, \mathbf{Y}) q(\mathbf{f}_k^*) q(\mathbf{W}_k^*) d\mathbf{y}_p^* d\mathbf{W}_{k,p}^* d\mathbf{f}_k^* \\
&= \sum_{k=1}^K \pi_k \sigma_{\mathbf{y}_p}^2 I + \sum_{k=1}^K \pi_k \mathbb{E}_{q_k(\mathbf{f}^*, \mathbf{W}^*)} \left[\left(\sum_{q_1=1}^Q \mathbf{W}_{k,p,q_1}^* \mathbf{f}_{k,q_1}^* \right) \left(\sum_{q_2=1}^Q \mathbf{W}_{k,p,q_2}^* \mathbf{f}_{k,q_2}^* \right) \right]
\end{aligned} \tag{46}$$

In the case when $q_1 = q_2$ the expectation is

$$\begin{aligned}
&\sum_{q_1=1}^Q \mathbb{E}_{q_k(\mathbf{f}^*, \mathbf{W}^*)} [\mathbf{W}_{k,p,q}^* \mathbf{f}_{k,q}^* \mathbf{f}_{k,q}^* \mathbf{W}_{k,p,q}^*] \\
&= \sum_{q_1=1}^Q \mathbb{E}_{q_k(\mathbf{W}^*)} \left[\mathbf{W}_{k,p,q_1}^* \left(\boldsymbol{\mu}_{k,q_1}^{(f)}(\mathbf{x}^*) \boldsymbol{\mu}_{k,q_1}^{(f)}(\mathbf{x}^*) + \boldsymbol{\Sigma}_{k,q_1}^{(f)}(\mathbf{x}^*, \mathbf{x}^*) \right) \mathbf{W}_{k,p,q_1}^* \right] \\
&= \sum_{q_1=1}^Q \left[\boldsymbol{\mu}_{k,p,q_1}^{(W)}(\mathbf{x}^*) \boldsymbol{\mu}_{k,q_1}^{(f)}(\mathbf{x}^*) \boldsymbol{\mu}_{k,q_1}^{(f)}(\mathbf{x}^*) \boldsymbol{\mu}_{k,p,q_1}^{(W)}(\mathbf{x}^*) + \boldsymbol{\mu}_{k,p,q_1}^{(W)}(\mathbf{x}^*) \boldsymbol{\Sigma}_{k,q_1}^{(f)}(\mathbf{x}^*, \mathbf{x}^*) \boldsymbol{\mu}_{k,p,q_1}^{(W)}(\mathbf{x}^*) \right. \\
&\quad \left. + \boldsymbol{\mu}_{k,q_1}^{(f)}(\mathbf{x}^*) \boldsymbol{\Sigma}_{k,p,q_1}^{(W)}(\mathbf{x}^*, \mathbf{x}^*) \boldsymbol{\mu}_{k,q_1}^{(f)}(\mathbf{x}^*) + \boldsymbol{\Sigma}_{k,q_1}^{(f)}(\mathbf{x}^*, \mathbf{x}^*) \boldsymbol{\Sigma}_{k,p,q_1}^{(W)}(\mathbf{x}^*, \mathbf{x}^*) \right]
\end{aligned} \tag{47}$$

In all other cases there will be no square terms

$$\sum_{q_1=1}^Q \sum_{q_2 \neq q_1}^Q \mathbb{E}_{q_k(\mathbf{f}^*, \mathbf{W}^*)} [\mathbf{W}_{k,p,q_1}^* \mathbf{f}_{k,q_1}^* \mathbf{f}_{k,q_2}^* \mathbf{W}_{k,p,q_2}^*] = \sum_{q_1=1}^Q \sum_{q_2 \neq q_1}^Q \boldsymbol{\mu}_{k,p,q_1}^{(W)}(\mathbf{x}^*) \boldsymbol{\mu}_{k,q_1}^{(f)}(\mathbf{x}^*) \boldsymbol{\mu}_{k,q_2}^{(f)}(\mathbf{x}^*) \boldsymbol{\mu}_{k,p,q_2}^{(W)}(\mathbf{x}^*). \tag{48}$$

Combining both cases together we can write

$$\begin{aligned}
\mathbb{E}[(\mathbf{y}_p^*)^2] &= \sum_{k=1}^K \pi_k \sigma_{\mathbf{y}_p}^2 I + \sum_{k=1}^K \pi_k \sum_{q_1=1}^Q [\boldsymbol{\mu}_{k,p,q_1}^{(W)}(\mathbf{x}^*) \boldsymbol{\Sigma}_{k,q_1}^{(f)}(\mathbf{x}^*, \mathbf{x}^*) \boldsymbol{\mu}_{k,p,q_1}^{(W)}(\mathbf{x}^*) \\
&\quad + \boldsymbol{\mu}_{k,q_1}^{(f)}(\mathbf{x}^*) \boldsymbol{\Sigma}_{k,p,q_1}^{(W)}(\mathbf{x}^*, \mathbf{x}^*) \boldsymbol{\mu}_{k,q_1}^{(f)}(\mathbf{x}^*) + \boldsymbol{\Sigma}_{k,q_1}^{(f)}(\mathbf{x}^*, \mathbf{x}^*) \boldsymbol{\Sigma}_{k,p,q_1}^{(W)}(\mathbf{x}^*, \mathbf{x}^*)] \\
&\quad + \sum_{k=1}^K \pi_k \sum_{q_1=1}^Q \sum_{q_2=1}^Q \boldsymbol{\mu}_{k,p,q_1}^{(W)}(\mathbf{x}^*) \boldsymbol{\mu}_{k,q_1}^{(f)}(\mathbf{x}^*) \boldsymbol{\mu}_{k,q_2}^{(f)}(\mathbf{x}^*) \boldsymbol{\mu}_{k,p,q_2}^{(W)}(\mathbf{x}^*).
\end{aligned} \tag{49}$$

B Synthetic Examples

Apart from the variational experiments in Section 2 of the main paper, additional experiments using a Markov Chain Monte Carlo (MCMC) approach are conducted in this section. We show that when the dependency structure is lost through the product likelihood construction, the mean of the posterior distribution for the latent function will also deviate from the true one. We also demonstrate the posterior contraction and the effect of the different corrections.

B.1 Data Generating Process

We generate two synthetic observation processes from:

$$\begin{aligned}
y_i^{(1)} &= f(x_i) + \epsilon_1 \\
y_j^{(2)} &= \frac{1}{2} \sum_{k=2j-1}^{2j} y_k^{(1)} + \epsilon_2
\end{aligned} \tag{50}$$

Where $y_i^{(1)}, i = 1, \dots, 2N$ is the observed value of $f(x_i)$ with noise $\epsilon_1 \sim \mathcal{N}(0, \sigma_1^2)$ and $y_j^{(2)}, j = 1, \dots, N$ is the aggregate function of $y^{(1)}$ with noise $\epsilon_2 \sim \mathcal{N}(0, \sigma_2^2), \sigma_1 = 1, \sigma_2 = 0.1$. We are using a sin function to generate data $f(x_i) = 5 \sin^2(x_i)$. The likelihood function with the observation processes $\mathbf{Y}_1 = \{y_i^{(1)}\}_{i=1}^{2N}$, $\mathbf{Y}_2 = \{y_j^{(2)}\}_{j=1}^N$ is given by:

$$L(\mathbf{Y}_1, \mathbf{Y}_2) = p(\mathbf{Y}_1 | \mathbf{f}(\mathbf{x}), \sigma_1^2) p(\mathbf{Y}_2 | \mathbf{Y}_1, \sigma_2^2) \tag{51}$$

When the data from \mathbf{Y}_1 has the same support as the observation process \mathbf{Y}_2 , the evidence from \mathbf{Y}_2 will not affect parameter estimation in the probability function $p(\mathbf{Y}_1 | \mathbf{f}(\mathbf{x}), \sigma_1)$. However, when \mathbf{Y}_2 has different support from the observed \mathbf{Y}_1 , the additional evidence should impact parameter inference. As \mathbf{Y}_2 does not depend on the latent function, this evidence will be hard to pass via the likelihood function in Eq. 51. One way to correct for this is to introduce dependency between \mathbf{Y}_1 and \mathbf{Y}_2 through a non-parametric prior over the latent function $\mathbf{f}(\mathbf{x})$.

B.2 Gaussian Processes: Product Likelihood

Since the two observation processes follow the same underlying function $\sin^2(x)$, we use a single Gaussian process to model the latent function $f(x)$. We assume:

$$f(x) \sim \mathcal{GP}(0, k(x, x')) \tag{52}$$

where $k(x, x')$ is the covariance function of $f(x)$. We are using the squared exponential kernel:

$$k(x, x') = A \exp\left(-\frac{(x - x')^2}{l}\right) \tag{53}$$

where A is the amplitude parameter and l is the length scale for the kernel function. Thus, we can write down the joint distribution of \mathbf{Y}_1 and \mathbf{Y}_2 as:

$$p(\mathbf{Y}_1, \mathbf{Y}_2, \mathbf{f}(\mathbf{x})) = p(\mathbf{f}(\mathbf{x}) | \theta) p(\mathbf{Y}_1 | \mathbf{f}(\mathbf{x}), \sigma_1^2) p(\mathbf{Y}_2 | \mathbf{Y}_1, \sigma_2^2) \tag{54}$$

Where θ is the hyper-parameters for the Gaussian process. We can write down the distribution of \mathbf{Y}_1 and \mathbf{Y}_2 by marginalizing out the latent function:

$$p(\mathbf{Y}_1, \mathbf{Y}_2) = \int p(\mathbf{f}(\mathbf{x})|\theta)p(\mathbf{Y}_1|\mathbf{f}(\mathbf{x}), \sigma_1^2)p(\mathbf{Y}_2|\mathbf{Y}_1, \sigma_2^2)d\mathbf{f}(\mathbf{x}) \quad (55)$$

As \mathbf{Y}_1 has all the information of $\mathbf{f}(\mathbf{x})$, this integral is tractable and we can write $y^{(1)} \sim \mathcal{GP}(0, k(x, x') + \sigma_1)$. But when the aggregation function \mathbf{Y}_2 has additional information about the latent function, i.e. \mathbf{Y}_1 and \mathbf{Y}_2 only partially overlapping, bringing additional information from \mathbf{Y}_2 requires the prediction of the missing values of the corresponding \mathbf{Y}_1 process. This can be done in Markov Chain Monte Carlo(MCMC) setting by treating the unobserved value of \mathbf{Y}_1 as extra parameters. However, this increase a lot of computational complexity for the MCMC sampler. One option is to make an independence assumption for \mathbf{Y}_1 and \mathbf{Y}_2 . Thus, the information in \mathbf{Y}_2 can affect the latent function \mathbf{f} directly.

B.3 Gaussian Processes: Composite Likelihood

Using the composite likelihoods, we assume each part of the likelihood is independent to each other. For the joint probability of \mathbf{Y}_1 and \mathbf{Y}_2 , we have:

$$p(\mathbf{Y}_1, \mathbf{Y}_2) = \int p(\mathbf{f}(\mathbf{x})|\theta)p(\mathbf{Y}_1|\mathbf{f}(\mathbf{x}), \sigma_1^2)p(\mathbf{Y}_2|\mathbf{f}(\mathbf{x}), \sigma_2^2)d\mathbf{f}(\mathbf{x}) \quad (56)$$

Instead of assuming the conditional probability $p(\mathbf{Y}_2|\mathbf{Y}_1, \sigma_2^2)$, we are now assuming the data depends on the latent function $\mathbf{f}(\mathbf{x})$ directly. However, when \mathbf{Y}_1 and \mathbf{Y}_2 are different resolutions under the same support, this likelihood misspecifies the correlation and will make the inference of $f(x)$ contract into the observed mean. While this contraction actually equals to an extra bias to the data in the overlapping zone, the misspecified dependency structure will lead to an overfitting problem. This overfitting problem of product likelihoods has been studied in the information theory [31, 34] and the simplest way is to use an exponential weight to correct the inference:

$$L(\mathbf{Y}_1, \mathbf{Y}_2) = \int p(\mathbf{f}(\mathbf{x})|\theta)p(\mathbf{Y}_1|\mathbf{f}(\mathbf{x}), \sigma_1^2)^\alpha p(\mathbf{Y}_2|\mathbf{f}(\mathbf{x}), \sigma_2^2)^\alpha d\mathbf{f}(\mathbf{x}) \quad (57)$$

where $\alpha \in \mathbb{R}_{>0}$ is composite weight for the likelihood. The problem of learning the latent function becomes learning the parameters of the likelihood function and the composite weights.

B.4 Composite Weights

The composite log likelihood function can be written as:

$$\ell_c(\hat{\theta}) = \sum_{i=1}^k f(\hat{\theta}_i|\mathbf{Y}) \quad (58)$$

where $f(\hat{\theta}_i|\mathbf{Y})$ is the likelihood function of i -th parameter θ_i and we assume each part of the likelihood function is independent to each other. $\hat{\theta}_i$ is the estimated value of θ_i . With the observed distribution of \mathbf{Y} , $p_0(\mathbf{Y}|\theta_0)$ and θ_0 as the true parameter value, we have:

$$\ell'_c(\theta_0) = \ell'_c(\hat{\theta}) + (\theta_0 - \hat{\theta})\ell''_c(\hat{\theta}) + o(n^{-1}) \quad (59)$$

$$\hat{\theta} - \theta_0 \rightarrow -\frac{\ell'(\theta_0)}{\ell''(\theta_0)} \quad (60)$$

Since we have $\ell'(\theta) = J(\theta)$ and $\ell''(\theta) = H(\theta)$, the variance of θ will follow the sandwich variance $H^{-1}(\theta)J(\theta)H^{-1}(\theta)$. Then, calculating the Taylor expansion for the likelihood, we have:

$$\ell_c(\theta_0) = \ell_c(\hat{\theta}) + (\theta_0 - \hat{\theta})\ell'_c(\hat{\theta}) + \frac{1}{2}(\theta_0 - \hat{\theta})\ell''_c(\hat{\theta})(\theta_0 - \hat{\theta})^T + o(n^{-1}) \quad (61)$$

The expected variance from the composite likelihood model is :

$$\mathbb{E}_\theta[Var(\hat{\theta}|\mathbf{Y})] = -H(\hat{\theta}|\mathbf{Y}) \quad (62)$$

Since $\hat{\theta} \rightarrow \theta$, we need to set the variance of the estimated parameter to the asymptotic variance. Thus, we have:

$$\mathbb{E}_\theta[\alpha Var(\hat{\theta}|\mathbf{Y})] = H^{-1}(\hat{\theta}|\mathbf{Y})J(\hat{\theta}|\mathbf{Y})H^{-1}(\hat{\theta}|\mathbf{Y}) \quad (63)$$

For a scalar variable, we can match the variance to the exact asymptotic variance using a scalar number. But if the estimating variable θ is high dimensional, it's not easy to adjust the proper variance using a single weight. We could use a matrix ($\mathbf{C} \in \mathbb{R}^{k \times k}$) to adjust the covariance structure. In this case we would have:

$$\mathbf{C}H(\hat{\theta}|\mathbf{Y})\mathbf{C}^T = H^{-1}(\hat{\theta}|\mathbf{Y})J(\hat{\theta}|\mathbf{Y})H^{-1}(\hat{\theta}|\mathbf{Y}) \quad (64)$$

However, this increases the computational complexity substantially. One alternative way is to use a scalar weight to match the identities of the covariance matrix. Lyddon et al. [17] and Ribatet [27] developed two different ways to adjust the identities of the covariance matrix:

$$\alpha_{\text{Ribatet}} = \frac{|\hat{\theta}|}{Tr[\mathbf{H}(\hat{\theta})^{-1}\mathbf{J}(\hat{\theta})]} \quad , \quad \alpha_{\text{Lyddon}} = \frac{Tr[\mathbf{H}(\hat{\theta})\mathbf{J}(\hat{\theta})^{-1}\mathbf{H}(\hat{\theta})]}{Tr[\mathbf{H}(\hat{\theta})]} \quad (65)$$

Where α_{Ribatet} considers all the information in the covariance matrix and α_{Lyddon} only matches the information in the diagonal elements.

B.5 MCMC Composite Likelihood Experiments

We now construct an MCMC experiment for the synthetic data using Eq. 50. Instead of sampling directly from the intractable joint distribution of $L(\mathbf{Y}_1, \mathbf{Y}_2)$, we sample from the joint probability with the latent variable $L(\mathbf{Y}_1, \mathbf{Y}_2, \mathbf{f}(\mathbf{x}))$ via a Metropolis-Hastings within Gibbs sampler. We perform three block updates: on θ_0 for the Gaussian process prior, $\mathbf{f}(\mathbf{x})$ for the latent function variables and $\sigma^2 = \{\sigma_1^2, \sigma_2^2\}$ for the noise parameter.

Algorithm 3 Block Metropolis-Hastings within Gibbs

Input: Observed datasets $\{(\mathbf{X}_s, \mathbf{Y}_s)\}_{s=1}^S$, initial parameters θ_0 ,

for i -th iteration **do**

 Update parameter block θ_i

function BLOCK (θ_i)

 1. Sample proposed value of the Gaussian process prior $\theta'_i \sim N(\theta_{i-1}, \Delta)$

 2. Calculate the conditional probability distribution $p(\theta'_i|Y_1, Y_2, \sigma_{i-1}^2, f(x)_{i-1})$

 3. Calculate the acceptance rejection ratio:

$$\pi = \frac{p(\theta'_i|Y_1, Y_2, \sigma_{i-1}^2, \mathbf{f}(\mathbf{x})_{i-1})}{p(\theta_{i-1}|Y_1, Y_2, \sigma_{i-1}^2, \mathbf{f}(\mathbf{x})_{i-1})}$$

 4. Update the i -th value of θ_i via π

end function

 Update the parameter block $\mathbf{f}(\mathbf{x})_i$ via $\pi = \frac{p(\mathbf{f}(\mathbf{x})'_i|Y_1, Y_2, \sigma_{i-1}^2, \theta_i)}{p(\mathbf{f}(\mathbf{x})_{i-1}|Y_1, Y_2, \sigma_{i-1}^2, \theta_i)}$

 Update the parameter block σ_i via $\pi = \frac{p(\sigma_i'^2|Y_1, Y_2, \theta_i, \mathbf{f}(\mathbf{x})_i)}{p(\sigma_{i-1}^2|Y_1, Y_2, \theta_i, \mathbf{f}(\mathbf{x})_i)}$

end for

return $\theta, \mathbf{f}(\mathbf{x}), \sigma^2$

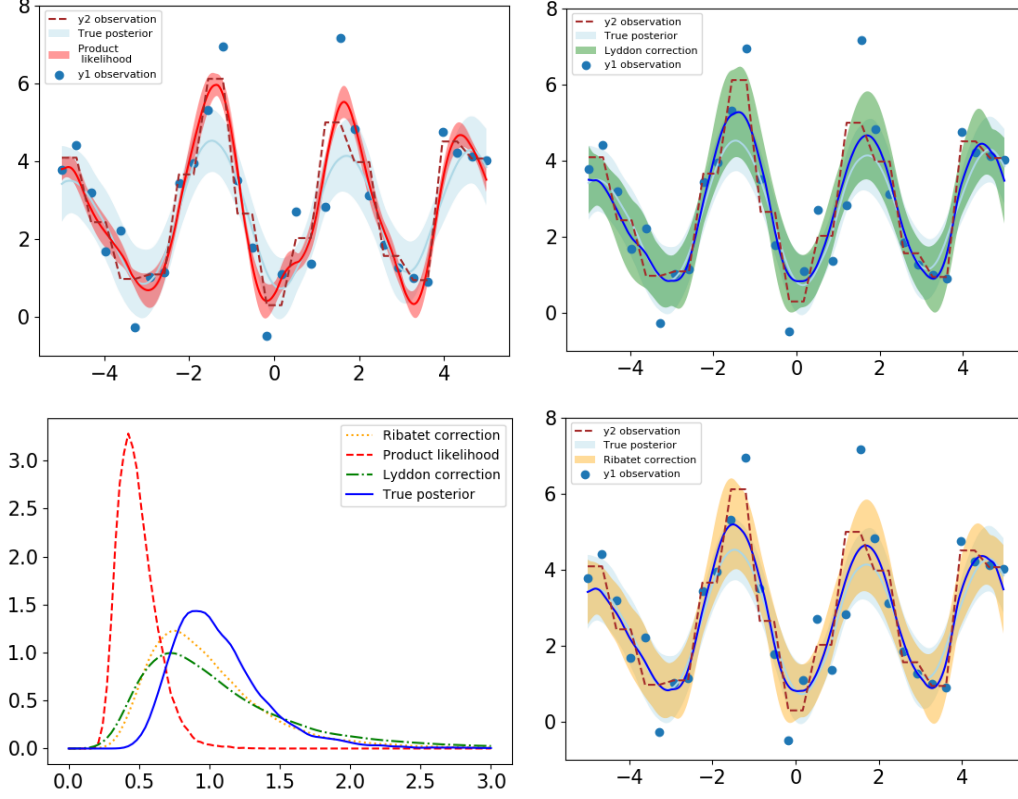


Figure 5: **Top Left:** Comparing the posterior of the latent function $f(x)$ under the product likelihood assumption and a correctly specified likelihood. The product likelihood assumption causes extreme posterior contraction which effects both the mean and variance. **Bottom Left:** Comparison of the true posterior of σ_1 noise to the posterior under the product likelihood and under the composite likelihood with different weights. The composite likelihood is able to recover the true posterior. **Top Right:** Comparing the posterior of the latent function $f(x)$ under the composite likelihood assumption with Lyddon correction and a correctly specified likelihood. **Bottom Right:** Comparing the posterior of the latent function $f(x)$ under the composite likelihood assumption with Ribatet correction and a correctly specified likelihood. The two correction have the similar results for our experiments. Although the mean of the function is not exactly match the true function, the variance of the latent function is corrected close to the true function. The misspecified part is due to the imperfect match of the asymptotic variance discussed in section B.4

B.6 Variational Composite Likelihood Experiments

In this section we provide further details to reproduce the variational composite likelihood experiments in Sec. 2.2 and Fig. 2 of the main paper.

Data Generation We consider the case of having two dependent observation processes. We generate one process $\mathbf{Y}_1 = 5 \cdot \sin(\mathbf{X})^2 + 0.1 \cdot \epsilon$ with $\epsilon \sim \mathcal{N}(0, 1)$ with 100 samples over the range $[-2, 15]$. For \mathbf{Y}_2 we aggregate \mathbf{Y}_1 into bins of size 3, $\mathcal{S}_2 = 3$, so that $\mathbf{Y}_2 \in \mathbb{R}^{33}$ and $\mathbf{X}_2 \in \mathbb{R}^{33 \times 3}$. In Fig. 2 we only plot the range $[3, 10]$.

Parameter Initialization For both MR-GP and VBAGG-NORMAL we use an SE kernels with length-scale of 0.1 and variance 1.0. We initialize the likelihood noise to 0.1.

Additional Training Details For VBAGG-NORMAL we run for 5000 epochs. For MR-GP for we run the MLE estimate for 5000 epochs, obtain α_{Ribatet} and then optimize the ELBO for 5000 epochs.

C Multi-resolution Air Pollution Experiments

C.1 Inter-task Multi-resolution: PM_{10} - PM_{25}

In this section we provide additional details for reproducing the inter-task multi-resolution experiments as described in the main paper.

Variational Parameter Initialization: For MR-DGP we initialize all likelihood noises to 0.01 and we use a Matern32 kernel for all latent functions with a lengthscale of 0.01. For both MR-GPRN and CENTER-POINT we initialize the likelihood noise to be 0.1 and use a squared exponential kernel for all latent functions. We use $Q = 1$ and set the lengthscale of \mathbf{f} to be 0.1 and the lengthscales of \mathbf{W} to be 3.0.

Training Details: We train MR-DGP for a total of 2000 iterations. We train both MR-GPRN and CENTER-POINT for 2000 iterations each.

C.2 Intra-task Multi-resolution: Space-time NO_2

In this section we provide additional details for reproducing the *intra*-task multi-resolution experiments described in Sec. 4 of the main paper.

Data pre-processing: We extract spatial features based on the London road network (OS Highways)² and land use (UKMap)³. OS Highways is a dataset of every road in London with information of the length, road classification (A Road, B Road, etc). UKMap is a dataset of polygons where each polygon represents a physical entity, e.g a building, a river, a park, etc. UKMap provides additional information such as the height of the buildings and the area of the parks and rivers. For each input location we construct a buffer of approximately 100m (a radius 0.001 degrees in SRID:4326). Within the buffer zone we calculate the average length of the A-roads, the average ratio between the width of the roads and height of buildings on the corresponding roads, and the total area of vegetation and water. We convert all time stamps into unix epochs and we standardize all features before training. To approximate the integral in the likelihood (Eq. 4 in main text) we discretize the area of each satellite based observation input into a 10 by 10 uniform grid of lat-lon points.

MR-DGP Architecture: For MR-DGP we use the architecture described on the right subfigure of Fig. 3 in the main paper where $\mathbf{X}_2, \mathbf{Y}_2$ corresponds to the LAQN dataset and $\mathbf{X}_1, \mathbf{Y}_1$ to the satellite dataset. We give the initialization of the specific latent functions below.

Variational Parameter Initialization: For MR-GPRN and VBAGG-NORMAL we use 400 inducing points for all latent functions. Both the inducing function values and the variances are randomly initialized between 0 and 1. For MR-DGP the latent functions $\mathbf{f}_{2,2}$ and $\mathbf{f}_{1,1}$ we place 200 inducing points and for \mathbf{f}_2 we use 100. For all models we initialize the inducing points locations with K-means.

Model Parameter Initialization: In all models and latent function withing, MR-GPRN, VBAGG-NORMAL and MR-DGP we use SE kernels initialized with lengthscales of 0.1 and SE variance to 1.0. We initialize the likelihood noise to be 0.1.

Additional Training Details: We train MR-DGP for a total of 9000 iterations. We train both MR-GPRN, VBAGG-NORMAL and CENTER-POINT for 10000 iterations each.

D MR-DGP

In this section we provide the complete derivation of the variational lower bound for MR-DGP.

²<https://www.ordnancesurvey.co.uk/business-and-government/help-and-support/products/os-mastermap-highways-network.html>

³<https://www.geoinformationgroup.co.uk/ukmap>

D.1 Specification of the prior

The full prior for MR-DGP is given by

$$p(\mathbf{F}, \mathbf{M}) = \left(\prod_{p=2}^P p(\mathbf{f}_p | \mathbf{m}_p) p(\mathbf{m}_p | \mathbf{f}_{1,p}, \{\mathbf{f}_{a,p}^{(2)}\}_{a=2}^A) \right) \left(\prod_{p=1}^P p(\mathbf{f}_{1,p}) \prod_{a=2}^A p(\mathbf{f}_{a,p}^{(2)} | \mathbf{f}_{a,p}) p(\mathbf{f}_{a,p}) \right) \quad (66)$$

We say that each of the GPs $\mathbf{f}_{a,p}$ are base GPs and to simplify notation in subsequent sections we use the function $\text{Pa}(\cdot)$ to denote the set of parent functions for each node. For example, $\text{Pa}(\mathbf{m}_2) = \{\mathbf{f}_{1,p}\} \cup \{\mathbf{f}_{a,p}^{(2)}\}$.

D.2 Likelihood

The likelihood for MR-DGP (as illustrated in in Fig. 3) is

$$p(\mathbf{Y} | \mathbf{F}) = \prod_{p=2}^P p(\mathbf{Y}_{1,1} | \mathbf{f}_p) \prod_{p=1}^P p(\mathbf{Y}_{1,p} | \mathbf{f}_{a,p}) \prod_{a=2}^A p(\mathbf{Y}_{1,p} | \mathbf{f}_{a,1}^{(2)}) p(\mathbf{Y}_{a,p} | \mathbf{f}_{a,p}) \quad (67)$$

where each likelihood component is a multi-resolution likelihood of the form

$$\mathcal{N}\left(\mathbf{Y}_{a,n} \mid \frac{1}{|S_a|} \int_{S_{a,n}} \mathbf{f}_a^{(k)}(\mathbf{X}) d\mathbf{X}, \sigma_{a,k}^2 \mathbf{I}\right), \quad (68)$$

again we discretise the integral with a uniform grid over S_a .

D.3 Augmented Prior

To allow for efficient inference we sparsify each GP by introducing inducing points:

$$\begin{aligned} p(\mathbf{F}, \mathbf{M}, \mathbf{U}) &= p(\mathbf{F}, \mathbf{M} | \mathbf{U}) p(\mathbf{U}) \\ &= \left(\prod_{p=2}^P p(\mathbf{f}_p | \mathbf{m}_p, \mathbf{u}_p) p(\mathbf{m}_p | \text{Pa}(\mathbf{m}_p)) \right) \cdot \\ &\quad \left(\prod_{p=1}^P p(\mathbf{f}_{1,p} | \mathbf{u}_{1,p}) \prod_{a=2}^A p(\mathbf{f}_{a,p}^{(2)} | \mathbf{f}_{a,p}, \mathbf{u}_{a,p}^{(2)}) p(\mathbf{f}_{a,p} | \mathbf{u}_{a,p}) \right) \cdot \\ &\quad \left(\prod_{p=2}^P p(\mathbf{u}_p) \cdot \prod_{p=1}^P p(\mathbf{u}_{1,1}) \prod_{a=2}^A p(\mathbf{u}_{a,p}^{(2)}) p(\mathbf{u}_{a,p}) \right) \end{aligned} \quad (69)$$

where each $p(\mathbf{u}_{\cdot,\cdot}^{(\cdot)}) = \mathcal{N}(\mathbf{u}_{\cdot,\cdot}^{(\cdot)} | 0, \mathbf{K}_{\cdot,\cdot}^{(\cdot)}(\mathbf{Z}_{\cdot,\cdot}^{(\cdot)}, \mathbf{Z}_{\cdot,\cdot}^{(\cdot)}))$ for $\mathbf{u}_{\cdot,\cdot}^{(\cdot)} \in \mathbb{R}^M$. The locations of the inducing points for the base GPs are $\mathbf{Z}_{a,p} \in \mathbb{R}^{M \times D}$ and for the deep GPs $\mathbf{Z}_p, \mathbf{Z}_{a,p}^2 \in \mathbb{R}^{M \times D}$. For brevity we have omitted the conditional on the inducing locations \mathbf{Z} in our notation.

D.4 Variational approximate Posterior

Following [28] we construct an approximate augmented posterior that maintains the dependency structure between layers:

$$\begin{aligned} q(\mathbf{M}, \mathbf{F}, \mathbf{U}) &= p(\mathbf{M}, \mathbf{F} | \mathbf{U}) q(\mathbf{U}) \\ &= p(\mathbf{M}, \mathbf{F} | \mathbf{U}) \prod_{p=2}^P q(\mathbf{u}_p) \cdot \prod_{p=1}^P q(\mathbf{u}_{1,1}) \prod_{a=2}^A q(\mathbf{u}_{a,p}^{(2)}) q(\mathbf{u}_{a,p}) \end{aligned} \quad (70)$$

where each $q(\mathbf{u}_{\cdot,\cdot}^{(\cdot)})$ are standard free-form Gaussians $\mathcal{N}(\mathbf{u}_{\cdot,\cdot}^{(\cdot)} | \mathbf{m}_{\cdot,\cdot}^{(\cdot)}, \mathbf{S}_{\cdot,\cdot}^{(\cdot)})$. The conditional is of the form

$$p(\mathbf{f}_{a,p}|\text{Pa}(\mathbf{f}_{a,p}), \mathbf{u}_{a,p}) = \mathcal{N}(\mathbf{f}_{a,p} \mid \mu'_{a,p}, \Sigma'_{a,p}) \quad (71)$$

with mean and variance given by the standard conditional equations

$$\begin{aligned} \mu'_{a,p} &= \alpha_{a,p} \mathbf{K}_{a,p}(\mathbf{Z}_{a,p}, \mathbf{Z}_{a,p})^{-1} \mathbf{u}_{a,p}, \\ \Sigma'_{a,p} &= \mathbf{K}_{a,p}(\text{Pa}(\mathbf{f}_{a,p}), \text{Pa}(\mathbf{f}_{a,p})) - \alpha_{a,p} \mathbf{K}_{a,p}(\mathbf{Z}_{a,p}, \mathbf{Z}_{a,p})^{-1} \alpha_{a,p}^T. \end{aligned} \quad (72)$$

where $\alpha_{a,p} = \mathbf{K}_{a,p}(\text{Pa}(\mathbf{f}_{a,p}), \mathbf{Z}_{a,p})$.

D.5 Marginalisation over inducing points

Firstly we can marginalise the inducing variables analytically

$$\begin{aligned} q(\{\mathbf{m}_p\}_p^P, \{\mathbf{f}_{a,p}\}_{a,p=1}^{\mathcal{A},P}) &= \int q(\{\mathbf{m}_p\}_p^P, \{\mathbf{f}_{a,p}, \mathbf{u}_{a,p}\}_{a,p=1}^{\mathcal{A},P}) d\mathbf{u}_{1,1} \cdots d\mathbf{u}_{\mathcal{A},P} \\ &= \prod_{p=1}^P p(\mathbf{m}_p|\text{Pa}(\mathbf{m}_p)) \prod_{a=1}^{\mathcal{A}} \int p(\mathbf{f}_{a,p}|\text{Pa}(\mathbf{f}_{a,p}), \mathbf{u}_{a,p}) q(\mathbf{u}_{a,p}) d\mathbf{u}_{a,p} \\ &= \prod_{p=1}^P p(\mathbf{m}_p|\text{Pa}(\mathbf{m}_p)) \prod_{a=1}^{\mathcal{A}} q(\mathbf{f}_{a,p}|\text{Pa}(\mathbf{f}_{a,p})) \end{aligned} \quad (73)$$

The integral can be evaluated in closed form resulting in

$$q(\mathbf{f}_{a,p}|\text{Pa}(\mathbf{f}_{a,p})) = \mathcal{N}(\mathbf{f}_{a,p} \mid \mu_{a,p}, \Sigma_{a,p}) \quad (74)$$

where the mean and variance are given by

$$\begin{aligned} \mu_{a,p} &= \alpha_{a,p} \mathbf{K}_{a,p}(\mathbf{Z}_{a,p}, \mathbf{Z}_{a,p})^{-1} \mathbf{m}_{a,p}, \\ \Sigma_{a,p} &= \Sigma'_{a,p} - \alpha_{a,p} \mathbf{K}_{a,p}(\mathbf{Z}_{a,p}, \mathbf{Z}_{a,p})^{-1} \mathbf{S}_{a,p} \mathbf{K}_{a,p}(\mathbf{Z}_{a,p}, \mathbf{Z}_{a,p})^{-1} \alpha_{a,p}^T \end{aligned} \quad (75)$$

D.6 Mixture of Experts

We define $p(\mathbf{m}_p|\text{Pa}(\mathbf{m}_p))$ as a mixture of experts, that is a weighted combination of the local experts

$$p(\mathbf{m}_p|\text{Pa}(\mathbf{m}_p)) = \mathcal{N}\left(\mathbf{m}_p \mid \sum_a^{\mathcal{A}} \mathbf{w}_{a,p} \mu_{a,p}, \sum_a^{\mathcal{A}} \mathbf{w}_{a,p} \Sigma_{a,p} \mathbf{w}_{a,p}\right). \quad (76)$$

The weights $\mathbf{w}_{a,p}$ are application specific and we provide specific examples in Section D.9.

D.7 Marginalisation over layers

We follow the doubly stochastic framework of [28] and marginalise through the layers using Monte Carlo estimates. The first layer of GPs can be sampled from directly, these samples are then propagated through all the subsequent layers:

$$q(\mathbf{f}_{a,p}) = \int q(\mathbf{f}_{a,p}|\text{Pa}(\mathbf{f}_{a,p})) \prod_{l=1}^{\mathcal{L}-1} q(\text{Pa}^{(l)}(\mathbf{f}_{a,p})|\text{Pa}^{(l+1)}(\mathbf{f}_{a,p})) d\text{Pa}^{(1)}(\mathbf{f}_{a,p}) \cdots d\text{Pa}^{(\mathcal{L})}(\mathbf{f}_{a,p}), \quad (77)$$

where each $q(\mathbf{f}_{a,p}|\text{Pa}(\mathbf{f}_{a,p}))$ is a DGP of the form Eqn. 6.

D.8 Variational Lower Bound

In this section we provide the derivation of the variational lower bound for MR-DGP. The evidence lower bound (ELBO), which lower bounds the log marginal likelihood $\log p(\mathbf{Y}|\mathbf{X})$, is

$$\mathcal{L} = \mathbb{E}_{q(\mathbf{M}, \mathbf{F}, \mathbf{U})} \left[\log \frac{p(\mathbf{Y}, \mathbf{M}, \mathbf{F}, \mathbf{U})}{q(\mathbf{M}, \mathbf{F}, \mathbf{U})} \right] = \mathbb{E}_{q(\mathbf{M}, \mathbf{F}, \mathbf{U})} \left[\log \frac{p(\mathbf{Y}|\mathbf{F})p(\mathbf{M}, \mathbf{F}|\mathbf{U})p(\mathbf{U})}{p(\mathbf{M}, \mathbf{F}|\mathbf{U})q(\mathbf{U})} \right] \quad (78)$$

Cancelling the relevant terms inside the logarithm we get

$$\mathcal{L}_{\text{MR-DGP}} = \underbrace{\mathbb{E}_{q(\mathbf{M}, \mathbf{F}, \mathbf{U})} [\log p(\mathbf{Y}|\mathbf{F})]}_{\text{ELL}} + \underbrace{\mathbb{E}_{q(\mathbf{U})} \left[\log \frac{p(\mathbf{U})}{q(\mathbf{U})} \right]}_{\text{KL}}.$$

Note that we have slightly abused notation to keep the derivation clear and now provide the full expanded lower bound:

$$\text{ELL} = \sum_{p=2}^P \mathbb{E}_{q(\mathbf{f}_p)} [\log p(\mathbf{Y}_{1,1}|\mathbf{f}_p)] + \sum_{p=1}^P \sum_a^A \left[\mathbb{E}_{q(\mathbf{f}_{a,1}^{(2)})} \left[p(\mathbf{Y}_{1,p}|\mathbf{f}_{a,1}^{(2)}) \right] + \mathbb{E}_{q(\mathbf{f}_{a,p})} [p(\mathbf{Y}_{a,p}|\mathbf{f}_{a,p})] \right] \quad (79)$$

and

$$\text{KL} = \mathbb{E}_{q(\mathbf{U})} \left[\log \frac{\left(\prod_{p=2}^P p(\mathbf{u}_p) \cdot \prod_{p=1}^P p(\mathbf{u}_{1,1}) \prod_{a=2}^A p(\mathbf{u}_{a,p}^{(2)}) p(\mathbf{u}_{a,p}) \right)}{\left(\prod_{p=2}^P q(\mathbf{u}_p) \cdot \prod_{p=1}^P q(\mathbf{u}_{1,1}) \prod_{a=2}^A q(\mathbf{u}_{a,p}^{(2)}) q(\mathbf{u}_{a,p}) \right)} \right]. \quad (80)$$

The KL term can be computed in closed form because it is just the sum of KL terms between two Gaussians. The ELL term is approximated using the Monte Carlo estimates from marginalising through the layers.

D.9 Mixture of Experts Weights

In this section we provide a specific and intuitive example of weights used when combining the mixture of experts. We derive weights that naturally weigh the experts by the level of support provided. We assume that this is defined by the resolution of the base layers, where higher resolutions are ‘trusted’ more. We first derive the weights for two generic GPs \mathbf{f}_1 and \mathbf{f}_2 and will generalise and apply the weights to the MR-GPRN after.

To find the support of an expert we utilise the differential entropy of the GP (see [13]):

$$H(p(\mathbf{f}|\mathbf{D})) = \frac{1}{2} \log(\sigma_{\mathbf{f}|\mathbf{D}}^2) + \frac{1}{2}(\log(2\pi) + 1). \quad (81)$$

For our mixing weights we are not interested in the amount of information each GP has, just whether there is any. So we drop the 2nd term on the rhs of Eq. 81, which we denote by $I(p(\mathbf{f}|\mathbf{D}))$, and normalise the information to be between zero and one. One such function is simply:

$$\hat{I}(\mathbf{f}) = \frac{I(p(\mathbf{f}|\mathbf{D})) - \min(I(p(\mathbf{f}|\mathbf{D})))}{\max(I(p(\mathbf{f}|\mathbf{D}))) - \min(I(p(\mathbf{f}|\mathbf{D})))} \quad (82)$$

This is not the only normalisation that can be done, one could also use a sigmoid function such as $\tanh(\cdot)$. Now we want a function that weighs \mathbf{f}_2 down when \mathbf{f}_1 has information, inherently capturing that we trust \mathbf{f}_1 over \mathbf{f}_2 . One such function is the normalised information of \mathbf{f}_2 minus the joint information of \mathbf{f}_1 and \mathbf{f}_2 :

$$\hat{I}(\mathbf{f}_1) + \beta_2 \hat{I}(\mathbf{f}_2) = \hat{I}(\mathbf{f}_1) + \hat{I}(\mathbf{f}_2) - \hat{I}(\mathbf{f}_1, \mathbf{f}_2). \quad (83)$$

Because we have normalised the information to be within zero and one we approximate the the joint information with a hadamard product (approximating an XOR function where the value is one if they both have information else it is zero).

$$\begin{aligned}\beta_2 \hat{I}(\mathbf{f}_2) &= \hat{I}(\mathbf{f}_2) - \hat{I}(\mathbf{f}_1) \hat{I}(\mathbf{f}_2) \\ &= (1 - \hat{I}(\mathbf{f}_1)).\end{aligned}\tag{84}$$

The function β_2 now has maximum and minimum values in the range $[0, 1]$ and so we can directly use β_2 as our mixing weights. To make all the weights sum to one we define

$$\beta_1 = 1 - \beta_2 = \hat{I}(\mathbf{f}_1).\tag{85}$$

D.10 Combining arbitrary number of experts

We combine them in a hierarchical manner, in a similar way to [?], and construct a computational graph. Given P experts, the mixing weights are defined as

$$\mathbf{m} = \hat{I}(\mathbf{f}_1)\mathbf{f}_1 + (1 - \hat{I}(\mathbf{f}_1))(\hat{I}(\mathbf{f}_2)\mathbf{f}_2 + (1 - \hat{I}(\mathbf{f}_2))(\cdots + (1 - \hat{I}(\mathbf{f}_{P-1}))\mathbf{f}_P))\tag{86}$$

E Relation to VBAGg

In this section we show that MR-GPRN can be seen as a generalisation of VBAGG-NORMAL [15] from a single GP to a GPRN. In VBAGg each observation y^a is the aggregate output of some bag of items $\mathbf{x}^a = \{\mathbf{x}_i^a\}_{i=1}^{N_a}$. The likelihood of each bag is $y^a | \mathbf{x}^a \sim \mathcal{N}(y | \eta^a, \tau^a)$ where $\eta^a = \sum_{i=1}^{N_a} w_i^a \mu(\mathbf{x}_i^a)$ and μ is the mean of the latent process f . In MR-GPRN we are modelling the underlying process with the sum of products of GPs. Rewriting MR-GPRN using the notation of [15]: $\mu = \mathbf{W}\mathbf{f}$ and each dataset $\{\mathbf{X}_a, \mathbf{Y}_a\}_{a=1}^A$ directly corresponds to the observations and bag of items defined in VBAGG-NORMAL. Let $N_a = S_a$, and $\tau^a = \sigma_a^2$ and the composite weight $\alpha = 1$. The composite weight of value 1 is implicitly included in the model of VBAGG-NORMAL through the independence assumption. We assume a simple aggregation of the bag of items, although we note that is not necessary, so setting $w_i^a = \frac{1}{|S_a|}$ we obtain $y^a \sim \mathcal{N}(\sum_{i=1}^{N_a} w_i^a \mu(\mathbf{x}_i^a), \tau^a)$ which is MR-GPRN in the notation of [15]. VBAGG-NORMAL is then recovered when we use only one latent function (by setting \mathbf{W} to a constant value), by only considering the single task setting and by setting the composite weight to one.

The VBAGg model is defined by

$$y^a \sim p(y^a | \eta^a), \quad \eta^a = \sum_{i=1}^{N_a} w_i^a \eta_i^a = \sum_{i=1}^{N_a} w_i^a \Psi(f(x_i^a))\tag{87}$$

where y^a is the independent aggregate observations, w_i^a are fixed weights and f follows a Gaussian process. For the unobserved latent variables y_i^a , we assume:

$$y_i^a \sim p(y_i^a | \eta_i^a), \quad \eta_i^a = \Psi(f(x_i^a))\tag{88}$$

When the probabilities of $y^a \sim p(y^a | \eta^a)$ and $y_i^a \sim p(y_i^a | \eta_i^a)$ are Gaussian, the model is equivalent to VBAGG-NORMAL. We firstly consider the special case, when $\Psi(f(x_i^a)) = f(x_i^a)$. As $f(x_i^a) \sim \mathcal{GP}(0, K(x_i^a, \cdot))$, we could say that $y_i^a \sim \mathcal{N}(f(x_i^a), \sigma_i^a)$ still follows a Gaussian process and $w_i y_i^a \sim \mathcal{GP}(0, w_i^2 K(x_i^a, \cdot))$ with fixed weight w_i . Thus, due to the additive property of Gaussian processes, the aggregation function $y^a \sim \mathcal{GP}(0, \sum_i \sum_j w_i w_j K(x_i^a, \cdot))$. The covariance of different aggregated

observations is:

$$\begin{aligned}
Cov(y^a, y^b) &= Cov\left(\sum_{i=1}^{N_a} w_i^a y_i^a, \sum_{j=1}^{N_b} w_j^b y_j^b\right) \\
&= \mathbb{E}\left[\sum_{i=1}^{N_a} \sum_{j=1}^{N_b} w_i^a y_i^a w_j^b y_j^b\right] - \mathbb{E}\left[\sum_{i=1}^{N_a} w_i^a y_i^a\right] \mathbb{E}\left[\sum_{j=1}^{N_b} w_j^b y_j^b\right] \\
&= \mathbb{E}\left[\sum_{i=1}^{N_a} \sum_{j=1}^{N_b} w_i^a y_i^a w_j^b y_j^b\right] \\
&= \sum_{i=1}^{N_a} \sum_{j=1}^{N_b} w_i^a w_j^b K(x_i^a, x_j^b) \\
&= \tilde{K}(\mathbf{x}^a, \mathbf{x}^b)
\end{aligned} \tag{89}$$

Also, the covariance between $y_i^* = f(x_i^*)$ and y^a is given:

$$\begin{aligned}
Cov(y_i^*, y^a) &= Cov\left(y_i^*, \sum_{j=1}^{N_a} w_j^a y_j^a\right) \\
&= \sum_{j=1}^{N_a} w_j^a K(x_i^*, x_j^a) \\
&= \tilde{K}^*(x_i^*, \mathbf{x}^a)
\end{aligned} \tag{90}$$

Thus, the problem of learning $f(\cdot)$ function becomes a standard Gaussian process regression problem and we have:

$$\begin{bmatrix} \mathbf{y} \\ y_i^* \end{bmatrix} \sim N\left(0, \begin{bmatrix} \tilde{K}(\mathbf{x}, \mathbf{x}) + \sigma_y^2 \mathbf{I} & \tilde{K}^*(x_i^*, \mathbf{x}) \\ \tilde{K}^*(\mathbf{x}, x_i^*) & K(x_i^*, x_i^*) \end{bmatrix}\right) \tag{91}$$

where $\mathbf{y} = \{y^1, \dots, y^a, \dots, y^n\}$ and the predictive posterior is given by:

$$\begin{aligned}
y_i^* | \mathbf{y}, \mathbf{x}, x_i^* &\sim N(\mu^*, \sigma^*) \\
\mu^* &= \tilde{K}^*(x_i^*, \mathbf{x}) [\tilde{K}(\mathbf{x}, \mathbf{x}) + \sigma_y^2 \mathbf{I}]^{-1} \mathbf{y} \\
\sigma^* &= K(x_i^*, x_i^*) - \tilde{K}^*(x_i^*, \mathbf{x}) [\tilde{K}(\mathbf{x}, \mathbf{x}) + \sigma_y^2 \mathbf{I}]^{-1} \tilde{K}^*(\mathbf{x}, x_i^*)
\end{aligned} \tag{92}$$

For a given generalized linear function $\Psi(f(x_i^a))$, we have $\Psi(\mathbf{f}(\mathbf{x}^a)) = \mathbf{W}_\Psi^a \mathbf{f}(\mathbf{x}^a)$, where \mathbf{W}_Ψ^a is a $N_a \times N_a$ weighting matrix. When $\mathbf{f}(\mathbf{x}^a) \sim \mathcal{GP}(0, K(x_i^a, x_j^a))$, then $\Psi(\mathbf{f}(\mathbf{x}^a)) = \mathbf{W}_\Psi^a \mathbf{f}(\mathbf{x}^a) \sim \mathcal{GP}(0, \mathbf{W}_\Psi^a K(x_i^a, x_j^a) (\mathbf{W}_\Psi^a)^T)$. Thus, we have:

$$\begin{aligned}
y^a &= \mathbf{w}^a [\Psi(\mathbf{f}(\mathbf{x}^a))]^T \\
&= \mathbf{w}^a [\mathbf{W}_\Psi^a \mathbf{f}(\mathbf{x}^a)]^T \sim \mathcal{GP}(0, \mathbf{w}^a [\mathbf{W}_\Psi^a \mathbf{K}(\mathbf{x}^a, \mathbf{x}^a) (\mathbf{W}_\Psi^a)^T] (\mathbf{w}^a)^T)
\end{aligned} \tag{93}$$

Following the Eq. 89, the covariance is given by:

$$Cov(y^a, y^b) = Cov(\mathbf{w}^a [\mathbf{W}_\Psi^a \mathbf{f}(\mathbf{x}^a)]^T, \mathbf{w}^b [\mathbf{W}_\Psi^b \mathbf{f}(\mathbf{x}^b)]^T) \sim \mathcal{GP}(0, \mathbf{w}^a [\mathbf{W}_\Psi^a \mathbf{K}(\mathbf{x}^a, \mathbf{x}^b) (\mathbf{W}_\Psi^b)^T] (\mathbf{w}^b)^T) \tag{94}$$

When the function $\Psi(\cdot)$ is a fixed function and \mathbf{W}_Ψ are constants we recover the VBAGG-NORMAL model. When the function $\Psi(\cdot)$ is a random function and \mathbf{W}_Ψ are Gaussian processes, then $\Psi(\mathbf{f}(\mathbf{x}^a))$ is a GPRN and the aggregation function y^a follows the MR-GPRN model without composite likelihood corrections.

References

- [1] (2008). Gaussian process product models for nonparametric nonstationarity. In *Proceedings of the 25th International Conference on Machine Learning*.

- [2] Adelsberg, M. and Schwantes, C. (2018). Binned kernels for anomaly detection in multi-timescale data using Gaussian processes. In *Proceedings of the KDD 2017: Workshop on Anomaly Detection in Finance*, Proceedings of Machine Learning Research.
- [3] Alvarez, M. A., Rosasco, L., Lawrence, N. D., et al. (2012). Kernels for vector-valued functions: A review. *Foundations and Trends® in Machine Learning*, 4(3):195–266.
- [4] Cutajar, K., Pullin, M., Damianou, A., Lawrence, N., and González, J. (2019). Deep Gaussian Processes for Multi-fidelity Modeling. *arXiv e-prints*, page arXiv:1903.07320.
- [5] Damianou, A. and Lawrence, N. (2013). Deep Gaussian processes. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*.
- [6] Deisenroth, M. P. and Ng, J. W. (2015). Distributed gaussian processes. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37, ICML’15*, pages 1481–1490. JMLR.org.
- [7] Fox, E. B. and Dunson, D. B. (2012). Multiresolution Gaussian processes. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*.
- [8] Fuentes, M. and Raftery, A. E. (2005). Model evaluation and spatial interpolation by Bayesian combination of observations with outputs from numerical models. *Biometrics*.
- [9] Gelfand, A., Fuentes, M., Guttorp, P., and Diggle, P. (2010). *Handbook of Spatial Statistics*. Chapman & Hall/CRC Handbooks of Modern Statistical Methods. Taylor & Francis.
- [10] Hensman, J., Fusi, N., and Lawrence, N. D. (2013). Gaussian processes for big data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*.
- [11] Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In *International Conference for Learning Representations*.
- [12] Knoblauch, J., Jewson, J., and Damoulas, T. (2019). Generalized Variational Inference. *arXiv e-prints*, page arXiv:1904.02063.
- [13] Krause, A., Singh, A., and Guestrin, C. (2008). Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies.
- [14] Krauth, K., Bonilla, E. V., Cutajar, K., and Filippone, M. (2017). AutoGP: Exploring the Capabilities and Limitations of Gaussian Process Models. In *Conference on Uncertainty in Artificial Intelligence (UAI)*.
- [15] Law, H. C. L., Sejdinovic, D., Cameron, E., Lucas, T. C., Flaxman, S., Battle, K., and Fukumizu, K. (2018). Variational learning on aggregate outputs with Gaussian processes. *Advances in Neural Information Processing Systems (NeurIPS)*.
- [16] Lázaro-Gredilla, M. and Titsias, M. K. (2011). Variational heteroscedastic Gaussian process regression. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*.
- [17] Lyddon, S. P., Holmes, C. C., and Walker, S. G. (2019). General Bayesian updating and the loss-likelihood Bootstrap. *Biometrika*.
- [18] Marécal, V., Peuch, V.-H., Andersson, C., Andersson, S., Arteta, J., Beekmann, M., Benedictow, A., Bergström, R., Bessagnet, B., Cansado, A., Chéroux, F., Colette, A., Coman, A., Curier, R. L., Denier van der Gon, H. A. C., Drouin, A., Elbern, H., Emili, E., Engelen, R. J., Eskes, H. J., Foret, G., Friese, E., Gauss, M., Giannaros, C., Guth, J., Joly, M., Jaumouillé, E., Josse, B., Kadygrov, N., Kaiser, J. W., Krajsek, K., Kuenen, J., Kumar, U., Liora, N., Lopez, E., Malherbe, L., Martinez, I., Melas, D., Meleux, F., Menut, L., Moinat, P., Morales, T., Parmentier, J., Piacentini, A., Plu, M., Poupkou, A., Queguiner, S., Robertson, L., Rouil, L., Schaap, M., Segers, A., Sofiev, M., Tarasson, L., Thomas, M., Timmermans, R., Valdebenito, A., van Velthoven, P., van Versendaal, R., Vira, J., and Ung, A. (2015). A regional air quality forecasting system over europe: the macc-ii daily ensemble production. *Geoscientific Model Development*.

- [19] Moreno-Muñoz, P., Artés-Rodríguez, A., and Álvarez, M. A. (2018). Heterogeneous multi-output Gaussian process prediction. In *Proceedings of the 32Nd International Conference on Neural Information Processing Systems*.
- [20] Nguyen, T. and Bonilla, E. (2013). Efficient variational inference for Gaussian process regression networks. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*.
- [21] Nguyen, T. and Bonilla, E. (2014a). Fast allocation of Gaussian process experts. In *Proceedings of the 31st International Conference on Machine Learning*.
- [22] Nguyen, T. V. and Bonilla, E. V. (2014b). Automated variational inference for Gaussian process models. In *Advances in Neural Information Processing Systems* 27.
- [23] Perdikaris, P., Raissi, M., Damianou, A., D. Lawrence, N., and Karniadakis, G. (2017). Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science*.
- [24] Perdikaris, P., Venturi, D., Royset, J. O., and Karniadakis, G. E. (2015). Multi-fidelity modelling via recursive co-kriging and Gaussian–markov random fields. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*.
- [25] Rasmussen, C. E. and Ghahramani, Z. (2002). Infinite mixtures of Gaussian process experts. In *Advances in Neural Information Processing Systems* 14.
- [26] Rasmussen, C. E. and Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press.
- [27] Ribatet, M. (2012). Bayesian inference from composite likelihoods, with an application to spatial extremes. In *Statistica Sinica* 22: 813–845.
- [28] Salimbeni, H. and Deisenroth, M. (2017). Doubly stochastic variational inference for deep Gaussian processes. In *Advances in Neural Information Processing Systems* 30.
- [29] Serban, I. V., Klinger, T., Tesauro, G., Talamadupula, K., Zhou, B., Bengio, Y., and Courville, A. (2017). Multiresolution recurrent neural networks: An application to dialogue response generation. In *Thirty-First AAAI Conference on Artificial Intelligence*.
- [30] Smith, M. T., Alvarez, M. A., and Lawrence, N. D. (2018). Gaussian process regression for binned data. *arXiv e-prints*.
- [31] Stoehr, J. and Friel, N. (2015). Calibration of conditional composite likelihood for Bayesian inference on Gibbs random fields. In *Artificial Intelligence and Statistics*.
- [32] Titsias, M. (2009). Variational learning of inducing variables in sparse Gaussian processes. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*.
- [33] Varin, C., Reid, N., and Firth, D. (2011a). An overview of composite likelihood methods. *Statist. Sinica*.
- [34] Varin, C., Reid, N., and Firth, D. (2011b). An overview of composite likelihood methods. *Statistica Sinica*.
- [35] Walder, C., Kim, K. I., and Schölkopf, B. (2008). Sparse multiscale Gaussian process regression. In *Proceedings of the 25th international conference on Machine learning*.
- [36] Wilson, A. G., Knowles, D. A., and Ghahramani, Z. (2012). Gaussian process regression networks. In *Proceedings of the 29th International Conference on Machine Learning*.
- [37] Yousefi, F., Smith, M. T., and Alvarez, M. A. (2019). Multi-task learning for aggregated data using gaussian processes. In *NeurIPS, 2019 (to appear)*.
- [38] Yuan, C. and Neubauer, C. (2009). Variational mixture of Gaussian process experts. In *Advances in Neural Information Processing Systems* 21.
- [39] Yusuke Tanaka, Toshiyuki Tanaka, T. I. T. K. M. O. Y. A. H. T. (2019). Spatially aggregated gaussian processes with multivariate areal outputs. In *NeurIPS, 2019 (to appear)*.